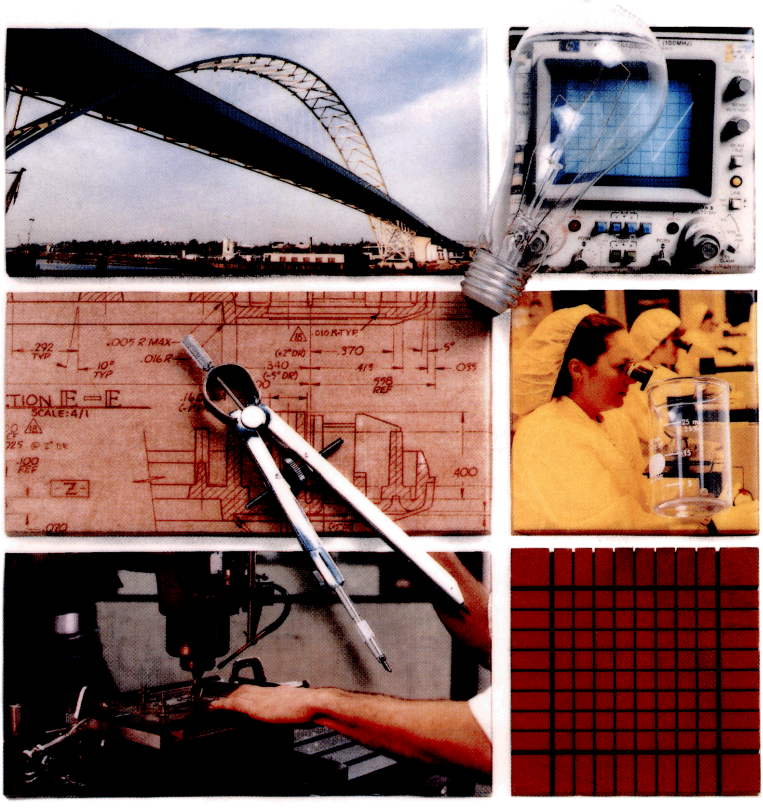
HEWLETT-PACKARD

HP-11C

BRUGERHÅNDBOG OG PROGRAMMERINGSVEJLEDNING



Bemærk

Hewlett-Packard A/S tilbyder ingen direkte eller indirekte garanti med hensyn til rigtigheden af tastsekvenser og programmer, der er optrykt i denne bog, ligesom der ikke garanteres for brugsværdi eller relevans for noget specielt formål. Tastsekvenserne og programmerne stilles til rådighed som de er, og hele risikoen med hensyn til deres kvalitet og anvendelighed ligger hos brugeren. Hvis tastsekvenser eller programmer viser sig at være fejlbehæftede, skal brugeren (og ikke Hewlett-packard eller nogen anden) bære alle omkostninger, der er forbundet med nødvendige korrektioner og udbedring af direkte skader og følgeskader. Hewlett-Packard A/S er ikke ansvarlig for nogen direkte skade eller følgeskade som opstår i forbindelse eller på grund af tilpasning, brug eller resultat af tastsekvenser eller programmer.

Hewlett-Packard A/S:
Datavej 52, 3460 Birkerød, tel. (02) 81 66 40
Navervej 1, 8600 Silkeborg, Tel. (06) 82 71 66

Hewlett-Packard Norge A/S:
Østerndalen 18, 1345 Østerås, tel. (02) 17 11 80

HP-11C

Brugerhåndbog
og
programmeringsvejledning

Maj 1981

00011-90018

Printed in Singapore

© Hewlett-Packard Company 1981

Introduktion

De vil hurtigt kunne bruge Deres HP-11C programmérbare lommeregner med kontinuerlig hukommelse. Denne brugerhåndbog giver Dem de oplysninger, der er nødvendige for at kunne bruge alle funktioner.
Hvis De aldrig før har arbejdet med en af Hewlett-Packards lommeregnere, bør De bruge lidt ekstra tid på afsnit 2, der i detaljer gennemgår den automatiske rullestak.
Under alle omstændigheder anbefales det, at De læser brugerhåndbogen igennem, løser opgaverne og regner med på eksemplerne. Det vil formodentlig være den hurtigste måde at opnå det bedst mulige kendskab til maskinen.
Brugerhåndbogen er opdelt i en indledning og tre kapitler, der behandler henholdsvis fastfunktioner, programmering og anvendelser.
Indledningen bør læses af alle; den giver et hurtigt overblik over nogle af de væsentligste muligheder.
Kapitel 1, »HP-11C som regnemaskine«, indeholder materiale, som erfarne HP-brugere kan læse kursorisk. Men alle bør gennemarbejde afsnit 1 »Grundbegreberne«, der opregner alle HP-11C-specifikke funktioner.
Stoffet i afsnittene 2-4 henvender sig først og fremmest til nye HP-brugere.
Kapitel 2 behandler programmeringen af HP-11C. Hvert afsnit handler om en særlig type programmeringsfunktioner, som først præsenteres ganske kort og derefter indarbejdes ved hjælp af større eksempler. De korte indledninger er let genkendelige: de er trykt på blå baggrund. De indeholder netop så megen information, at erfarne HP-brugere kun sjældent vil have brug for yderligere oplysninger.
Kapitel 3, afsnit 1 indeholder en række færdige programmer, som dækker et repræsentativt udsnit af HP-11C's anvendelsesområder. I afsnit 2 præsenteres en række programmeringstekniske ideer – en del af dem er behandlet med reference til programmer i programsamlingen.
I en række tillæg gøres der rede for bl.a. servicebetingelser, fejlmeddelelser, udskiftning af batterier og lignende praktiske forhold.
Endelig gives en kort oversigt over de enkelte funktioner.

Indhold

Introduktion 2
Indholdsfortegnelse 3
HP-11C's kontinuerlige hukommelse 9

HP-11C: En problemløser 10
Manuel problemløsning 11
Programmeret problemløsning 12

Kapitel 1 – HP-11C som regnemaskine

Afsnit 1: Grundbegreberne 15
Tænd/sluk – Afbryder 16
Lyspanel 16
Radixpunkt og 3-cifferinddeling 16
Indikatorer 16
Negative tal 17
Sletning og korrektion af tal i lyspanelet 17
Særlige udlæsningsformater 18
Overløb og underløb 18
Fejlmeddelelser 18
Batteriindikator 19
Hukommelse 19
Kontinuerlig hukommelse 19
Nulstilling af hele hukommelsen 19
Tastfunktioner 20
Primære og alternative funktioner 20
Annullering af forvalgstaster 21
Funktioner af én variabel 22
Funktioner af to variable 22
Indtastningsrækkefølge 22
Opgaver 22

Afsnit 2: Den automatiske rullestak, SIDSTE X og dataregistre

Den automatiske rullestak og stakmanipulation 25
Stakmanipulation 26
Funktioner og rullestakken 28
Funktioner af to variable 28
Kæderegning 30
SIDSTE X 31
Konstantaritmetik 33
Brug af dataregistre 35
Lagring af tal 35
Fremkald af tal 36
Nulstilling af dataregistre 36
Registeraritmetik 37
Opgaver 38

Afsnit 3: Matematiske funktioner

Pi 41
Ændring af udlæst tal 41
Fortegnsskift 41
Afrunding 41
Numerisk værdi 41
Heltalsdel 41
Decimaldel 41
Funktioner af én variabel 42
Generelle funktioner 42
Reciprokværdi 42
Fakultets- og gammafunktionen 42
Kvadratrod 43
Kvadrering 43
Trigonometriske funktioner 43
Vinkelmål 43
Trigonometriske funktioner 44
Omregning af tid og vinkler 44
Omregning mellem timer (grader) og timer/min/sek 45
Omregning mellem grader og radian 45
Logaritmefunktioner 46
Naturlig logaritme 46
Eksponentialfunktionen 46
10-talslogaritmen 46

Almindelig antilogaritme 46
Hyperbolske funktioner 46
Funktioner af to variable 47
Potensopløftning 47
Procent 47
Procentisk forskel 48
Omregning mellem polære og rektangulære koordinater 49
Sandsynlighedsregning 50
Permutationer 50
Kombinationer 50
Statistik 51
Pseudotilfældige tal 51
Summation af statistikdata 52
Korrektion af summerede data 55
Middelværdi 56
Standardafvigelse 57
Lineær regression 58
Lineære estimater og korrelationskoefficient 60

Afsnit 4: Udlæsningsformater

Udlæsningsformater 63
Fastkomma 64
Flydende tals repræsentation 64
Metrisk notation 65
Indtastning af eksponent 66
Mantisse 67
Afrunding af det 10. ciffer 68

Kapitel 2 – Programmering

Afsnit 1: Grundlæggende programmering

Hvad er et program? 70
Hvorfor skrive programmer? 70
Programkontrol 70
Automatisk splitning af hukommelsen 70
[MEM] 73
Tastkoder og linienumre 73
Forkortet indtastning 74
Funktioner til programkontrol 74
Statusskifter 74
Nulstilling af programhukommelsen 74

Hop til linie 000 74
Etiketter 74
Retur 75
Kør/Stop 75
Pause 75
USER-status 75
Programhukommelsen 76
Fortolkning af tastkoder 77
Programmering 79
Indledning og afslutning af et program 79
Det fuldstændige program 80
Lagring af programmet 80
Kørsel af programmet 81
Søgning efter etiketter 82
Udførelse af programinstruktioner 82
Ikke-programmérbare funktioner 83
Brug af USER-status 83
Stop og pauser under programkørsel 83
Programmerede stop 84
Programmerede pauser 86
Uventede programstop 88
Etiketter 89
Opgaver 89

Afsnit 2: Editering

Programfejl og kontinuerlig hukommelse 91
Fejlsøgning 91
Editeringsfunktioner 92
SST 92
BST 92
Hop til linie nnn 93
Slettetast 93
Eksempel på editering 93
-udførelse af et program 94
Brug af BST og SST i programmeringsstatus 95
Gennemsyn af det lagrede program 95
Hurtigt gennemsyn af program 96
Modificering af et program 96
Indsættelse af instruktioner 96

Hop til et bestemt linienummer 97
Forsøg på indsættelse af instruktioner i fuld program-hukommelse 99
Sletning af instruktioner 99
Opgaver 101

Afsnit 3: Forgreninger og kontroloperationer

Betingede operationer 105
Flag 106
Ubetinget hop 106
Forgreninger og løkker 107
Løkker 107
Betinget forgrening med løkkestruktur 108
Brug af flag 109

Afsnit 4: Underprogrammer

Kald af underprogram 113
Underprogrammer i flere niveauer 114
Brug af underprogrammer 115

Afsnit 5: Indeksregistret

I og i med forkortet indtastning 121
Indeksregistrets direkte funktioner 121
Lagring og fremkald af tallet i indeksregistret 121
Ombytning af X og I 121
Inkrementering og dekrementering af indeksregistret 122
Brug af ISG og DSE 123
Indeksregistrets indirekte funktioner 124
Indirekte lagring og fremkald 124
Indirekte ombytning med X 124
Indirekte registeraritmetik 124
Indirekte hop til etiket eller linie 125
Indirekte kald af underprogram 125
Eksempel på brug af løkke med ISG 126
Grænser for brug af ISG og DSE 128
Indirekte registeroperationer 128
Indirekte programkontrol 129
Indirekte adressering af etiketter 129
Indirekte adressering af linienumrene 130

Kapitel 3

Afsnit 1: Programsamling

Matrixoperationer 132
3 ligninger med 3 ubekendte 141
Trekantsberegninger 146
Kurvetilpasning 154
Newton-Raphson-iteration 160
U-bådsjagt 164
Finansregning. Annuiteter og rentes rente 175
t-test 184
X^2-test 190
Numerisk integration 194

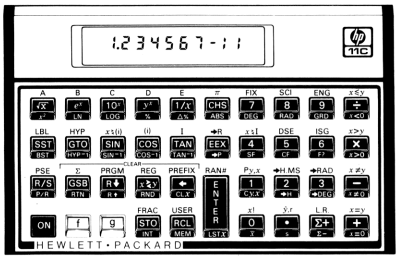
Afsnit 2: Programmeringsteknik

Struktur 199
Problemformulering 199
Algoritmen 200
Rutediagrammer 202
Underprogrammer 204
[f] [ISG] med [RCL] [i] 205
Indlæsning af data 206
Løkker 207
Flag 209
Tilfældige tal 210
Brugerdefinérbare taster 211

Tillæg A: Fejlmeddelelser 213
Tillæg B: Batteri, service og udbedring af mangler 217
Tillæg C: Automatisk splitning 225
Tillæg D: Stakløft og SIDSTE X 231
Programteknik – indeks 235
Oversigt over funktions- og programmeringstaster 238

Stikordsregister 245

HP11C's kontinuerlige hukommelse



AUTOMATISK RULLESTAK

T ➔
Z ➔
Y ➔
X ➔

Udlæses

SIDSTE X

PROGRAM MEMORY

Permanente	Fælles
000–	064–
001–	065–
002–	066–
061–	201–
062–	202–
063–	203–

STORAGE REGISTERS

Permanente		Fælles	
R_I			
R_0		$R_{.0}$	
R_1		$R_{.1}$	
R_2		$R_{.2}$	
R_3		$R_{.3}$	
R_4		$R_{.4}$	
R_5		$R_{.5}$	
R_6		$R_{.6}$	
R_7		$R_{.7}$	
R_8		$R_{.8}$	
R_9		$R_{.9}$	

Alle registerindhold bevares af den kontinuerlige hukommelse. Startsplitningen er på 63 programlinier og 20 dataregistre samt indeksregistret (R_I). HP-11C konverterer automatisk et dataregister til syv programlinier, efterhånden som der bliver behov for det. $R_{.9}$ er det første register, der konverteres, og R_0 er det sidste dataregister, der kan konverteres.

HP-11C
En problemløser

HP-11C programmérbar, teknisk lommeregner er en effektiv problemløser, der kan tages med næsten overalt. Den er egnet til løsning af alle typer opgaver lige fra de simpleste til de mest komplicerede – og dens kontinuerlige hukommelse kan bruges til at lagre vigtige data. HP-11C er så let at bruge og programmere, at der ikke kræves noget forudgående kendskab til programmering eller programmeringssprog for at kunne betjene den.

HP-11C har en indbygget batterisparer, der automatisk slukker for strømmen, når maskinen ikke har været brugt i 8-17 minutter. Men den kontinuerlige hukommelse bevarer al information, der er lagret i lommeregneren.

Hewlett-Packards lommeregnere er på mange punkter forskellige fra de fleste andre lommeregnere. En af de vigtigste forskelle ligger i indtastningssystemet, hvor vi har tasten [ENTER] i stedet for =. Styrken ved dette system bliver indlysende efter ganske kort tids brug. I et senere afsnit vil vi gennemgå HP's indtastningssystem i alle detaljer, så lige nu vil vi nøjes med at give et par eksempler på brugen af [ENTER].

HP-11C arbejder ikke med ventende operationer; svaret på en opgave udlæses så snart, der trykkes på en funktionstast. Lad os se på de aritmetiske funktioner:
[+] lægger det sidst indtastede tal til det tal, der allerede stod i maskinen. [−] trækker det fra, [×] ganger og [÷] dividerer. Operationerne foregår altid mellem det sidst indtastede tal og det tal, der i forvejen står i maskinen.
Men først skal vi have tal i maskinen. Man indtaster det første tal, trykker på [ENTER] for at skille det første fra det andet og indtaster så det andet tal. Når det er gjort, kan man trykke på [+], [−], [×] eller [÷] og få udført den pågældende operation mellem tallene.

Prøv nu at tænde for maskinen med [ON]
Hvis der står andet end nuller i lyspanelet skal De trykke på [←] . Hvis der ikke er fire nuller efter decimalpunktet, skal De trykke på [f] [FIX] 4, så tallene i lyspanelet kommer til at passe med de følgende eksempler. I denne håndbog bruges altid afrunding til 4 decimaler, undtagen i helt specielle tilfælde.

0.0000

Manuel problemløsning

Det er ikke nødvendigt at slette maskinen, før man starter på en ny opgave. Hvis man indtaster et forkert ciffer, kan det slettes med [←], hvorefter det korrekte ciffer kan indtastes.

Opgave	Indtastning	Lyspanel
9 + 6 = 15	9 [ENTER] 6 [+]	***15.0000***
9 − 6 = 3	9 [ENTER] 6 [−]	***3.0000***
9 × 6 = 54	9 [ENTER] 6 [×]	***54.0000***
9 ÷ 6 = 1,5	9 [ENTER] 6 [÷]	***1.5000***

Bemærk følgende, der gælder for alle fire eksempler:

- Begge tal er i maskinen, når der trykkes på [+], [−], [×] eller [÷].
- [ENTER] bruges kun til at adskille to tal, der indtastes lige efter hinanden.
- Resultatet bliver beregnet og udlæst, så snart man trykker på en funktionstast.

For at demonstrere den nære sammenhæng mellem manuel og programmeret problemløsning, vil vi først løse et problem manuelt – dvs. selv foretage alle indtastninger. Bagefter vil vi lave et program, der udfører den samme beregning.

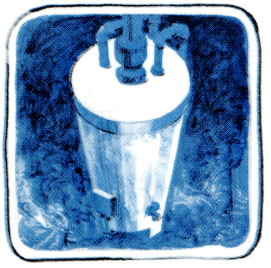
Opgaven går ud på at beregne varmetabet fra en cylindrisk varmtvandsbeholder. Varmetabet findes ved hjælp af udtrykket
$q = h \times A \times T$

Bemærkning: Når batterierne er næsten opbrugt, blinker en stjerne ★ i lyspanelets nederste, venstre hjørne. I tillæg D vises, hvordan man skifter batterier på HP-11C.

hvor

q = Varmetabet fra varmtvandsbeholderen
h = Varmeledningskoefficienten for beholderen
A = Beholderens overflade
T = Forskellen mellem beholderens og omgivelsernes temperatur

Eksempel: Varmtvandsbeholderen har en kapacitet på 200 liter.
Temperaturmålinger har vist, at beholderens overflade i gennemsnit er 8°C varmere end lufttemperaturen i rummet, hvor den er opstillet. Overfladearealet er 8 m^2, og varmeledningskoefficienten opgives til at være 2,67. Varmetabet i Watt kan nu beregnes således:



Indtastning	**Lyspanel**	
8 [ENTER]	***8.0000***	Indtastning af
8		temperaturforskellen T
		og overfladearealet A.
[×]	***64.0000***	Beregning af $T \times A$.
2.67	***2.67***	Varmeledningskoeffi-
		cienten h.
[×]	***170.8800***	Varmetabet i Watt.

Programmeret problemløsning

I det foregående eksempel blev varmetabet beregnet for en temperaturforskel på 8°C. Men hvad nu, hvis De skulle beregne varmetab ved en lang række temperaturforskelle? De kunne selvfølgelig udføre beregningen manuelt for hver enkelt temperatur – men der er en lettere måde: De kan skrive et program, hvor svaret fås efter indtastning af temperaturen og tryk på en enkelt tast.

Programmet skrives. Programmet består af de samme indtasatninger, som blev brugt til at løse opgaven manuelt. Dertil kommer to ekstra instruktioner: en etiket til at navngive programmet og en stopfunktion. De sættes henholdsvis før og efter de funktioner, der udgør selve programmet.

Lagring af programmet. Ved hjælp af tasterne på HP-11C kan De lagre et program i maskinens programhukommelse. Instruktionerne (dvs. etiket, beregnings- og stopfunktioner) registreres som programinstruktioner samtidig med, at de indtastes. I lyspanelet vises nogle koder, som vi senere vil komme tilbage til.

Indtastning	Lyspanel	
[g] [P/R]	*000–*	Sætter HP-11C i *programmeringsstatus* og viser PRGM i lyspanelet.
[f] CLEAR [PRGM]	*000–*	Sletter programhukommelsen.
[f] [LBL] [A]	*001–42,21,11*	Etiket A angiver starten på programmet.
8	*002– 8*	De samme taster, som blev brugt ved den manuelle beregning.
[×]	*003– 20*	
2	*004– 2*	
[.]	*005– 48*	
6	*006– 6*	
7	*007– 7*	
[×]	*008– 20*	
[g] [RTN]	*009– 43 32*	RTN afslutter programmet.
[g] [P/R]	*170.8800*	Ophæver programmeringsstatus; PRGM forsvinder, og det gamle resultat udlæses igen.

Kørsel af programmet. Programmet køres (udføres) ved at trykke på følgende taster:

Indtastning	Lyspanel	
8	*8*	Første temperaturforskel.
[f] [A]	*170.8800*	Varmetabet, som allerede er beregnet manuelt.
10 [f] [A]	*213.6000*	Varmetabet for en ny temperaturforskel.

Når programmet er skrevet og lagret, kan det bruges til lynhurtigt at beregne varmetabet for en lang række temperaturforskelle. Indtast temperaturforskellen og tryk [f] [A]. Udfyld tabellen til højre som kontrol af programmet.

Temp. forskel	Tab i Watt
5	?
7	?
9	?
11	?
13	?
15	?

Svarene skulle gerne blive: 106.8000, 149.5200, 192.2400, 234.9600, 277.6800, 320.4000.

Så let er det at programmere. Lommeregneren husker indtastningerne, og skal blot have at vide, hvor den skal starte, og hvor den skal slutte. Men der er selvfølgelig meget mere i det, end et så enkelt eksempel kan vise. Så lad os gå over til en grundig gennemgang af alle HP-11C's funktioner.

Ekstra programsamling

HP-11C Solutions Handbook. Denne håndbog fås som ekstraudstyr og indeholder en omfattende samling HP-11C-programmer inden for områderne matematik, statistik, elektronik, kemi, finans og spil.

HP-11C Solutions Handbook fås kun på engelsk og kan købes hos Deres HP-forhandler.

Kapitel 1

HP-11C
som regnemaskine

Afsnit 1

Grundbegreberne

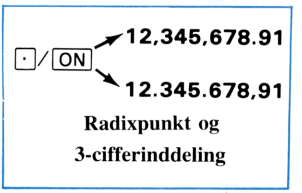
Tænd/Sluk – Afbryder

[ON] bruges til at tænde og slukke for HP-11C. For at spare på batterierne slukker lommeregneren automatisk, når den ikke har været brugt i 8-17 minutter.

Lyspanel

Radixpunkt og 3-cifferinddeling

Radixpunktet er det tegn, der bruges til at skille et tals heltalsdel og decimaldel. I Danmark (og de fleste andre europæiske lande) bruger man normalt komma som radixpunkt – i angelsaksiske lande bruges punktum. HP-11C er født med punktum som radixpunkt, men De kan vælge komma ved at slukke for maskinen og holde [·] nede, medens De igen tænder for maskinen.

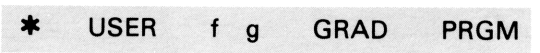


Radixpunkt og 3-cifferinddeling

I de følgende eksempler er der brugt komma som radixpunkt.

Indikatorer

Lyspanelet giver ikke blot numerisk information; det indeholder 6 indikatorer, der bruges til at angive maskinens status. Disse indikatorer beskrives i forbindelse med de funktioner og tilstande, som de fungerer sammen med.



Indikatorer i lyspanelet

Negative tal

Man kan skifte fortegn på tallet i lyspanelsregistret ved at trykke på fortegnsvenderen [CHS]. Når det udlæste tal repræsenterer et negativt tal, er der anbragt et minustegn foran tallets første ciffer. [CHS] kan også bruges til at fjerne dette minustegn, så tallet bliver positivt.

Sletning og korrektion af tal i lyspanelet: [CLx] [←]

HP-11C har to funktioner, der bruges til at rette eller slette indholdet af lyspanelet (eller mere præcist: lyspanelregistret): [CLx] *(sletning af x)* og [←] *(korrektion af sidste ciffer)*. Når [g] [CLx] bruges direkte fra tastaturet (dvs. når maskinen er i normalstatus), slettes eller nulstilles lyspanelregistret. Men bruges tasten, når maskinen er i programmeringsstatus, lagres [g] [CLx] som en instruktion.

[←] er en ikke-programmérbar funktion, der gør det muligt at slette lyspanelregistret i alle tilfælde.

1. Normalstatus

A. Under talindtastning sletter [←] kun det sidst indtastede *ciffer* og kan altså bruges til at rette et fejlindtastet tal, uden at man skal genindtaste hele tallet. Tasten kan bruges flere gange efter hinanden og kan på den måde bruges til nulstilling af hele lyspanelregistret.

Indtastning	**Lyspanel**	
12345	***12.345***	Når en talindtastning
[←]	***1.234***	ikke er afsluttet, sletter [←] kun det sidste
[←]	***123***	ciffer.
9	***1.239***	

B. [←] nulstiller hele lyspanelregistret efter stort set alle HP-11C's funktioner.

Næsten alle HP-11C's funktioner afslutter talindlæsning, hvorved lyspanelregistret skifter tilstand. Det medfører bl.a., at [←] skifter fra at slette det sidst indtastede ciffer til at slette hele tallet.

Indtastning	Lyspanel	
12345	***12.345***	Anvendes [←] efter brug af en funktion, slettes hele lyspanel-registret.
[√x]	***111,1081***	
[←]	***0,0000***	

2. I programmeringsstatus sletter [←] den programlinie, der udlæses.

Særlige udlæsningsformater

Lyspanelet bruges også til at give meddelelser fra maskinen til brugeren.

running. Medens et program er under udførelse og under beregning af *permutation/kombination* blinker lyspanelet »running«.

Overløb og underløb

Overløb. Hvis den numeriske værdi af tallet i lyspanelet bliver større end $9{,}999999999\times10^{99}$, udlæses overløbsmeddelelsen: alle 9-taller med det rigtige fortegn. Forekommer overløb under en programkørsel, standser programmet, og overløbsmeddelelsen vises i lyspanelet.

Underløb. Hvis et resultat er numerisk mindre end 1×10^{-99}, erstattes det af 0. Underløb stopper ikke udførelse af et program.

Fejlmeddelelser

Hvis man forsøger at udføre en beregning med et ugyldigt argument (som f.eks. kvadratroden af et negativt tal), vises en fejlmeddelelse i lyspanelet. I tillæg C finder De en fuldstændig liste over fejlmeddelelser og årsager til fejl. På bagsiden af HP-11C er der en kort liste over fejlmeddelelser.

Keystrokes	Display
4 [CHS]	**-4**
[√x]	**Error 0**
[←]	**-4.0000**

[←] bruges til at ophæve enhver fejltilstand. Man kan faktisk bruge en vilkårlig tast, men af hensyn til systematikken bør man vænne sig til at bruge [←].

Batteriindikator

Hvis en stjerne begynder at blinke i lyspanelets nederste venstre hjørne, betyder det, at der skal skiftes batterier. Se tillæg B.

** 0,0000*

Hukommelse

Kontinuerlig hukommelse

HP-11C's kontinuerlige hukommelse bevarer følgende data og statusinformation, selv når der slukkes for maskinen:

- Alle data, der er lagret i maskinen.
- Alle programmer, der er lagret i maskinen.
- Udlæsningsformat.
- Flagstatus.
- Splitning (forholdet mellem dataregistre og programregistre).
- Returadresser for ventende underprogrammer.
- Vinkelmål.

Når man tænder for HP-11C, starter den altid i normalstatus (PRGM vises *ikke* i lyspanelet) – også selvom den var i programmeringsstatus (PRGM vises), da der blev slukket for maskinen.

Den kontinuerlige hukommelse kan oven i købet bevare data og information, medens der skiftes batterier. Batteriskiftet skal foregå rimeligt hurtigt, men man taber altså ikke møjsommeligt indtastet information ved batteriskift.

Nulstilling af hele hukommelsen

Hvis De ønsker at nulstille alle data- og programregistre, kan det gøres på følgende måde:

1. Sluk for HP-11C
2. Tryk på [ON] og hold tasten nede.
3. Tryk på [−] og hold tasten nede.
4. Slip først [ON] og dernæst [−].

Nulstilling af hele hukommelsen

Når hele hukommelsen nulstilles vises en særlig fejlmeddelelse, der kan ophæves med [←]

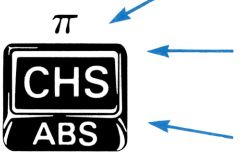
Pr Error

Bemærk: Man risikerer at nulstille hele hukommelsen, hvis maskinen tabes eller på anden måde udsættes for fysisk overlast.
Det samme sker, hvis batteriforbindelsen afbrydes i længere tid.

Tastfunktioner

Primære og alternative funktioner

De fleste af HP-11C's taster udfører én primær og to alternative funktioner. Den primære funktion, grundfunktionen, er altid markeret på tastens vandrette overflade. De alternative funktioner er markeret på tastaturpladen lige over tasten og på tastens skrå forside.



- Forvalg af en funktion, der er trykt med gult over en tast, udføres med den gule forvalgstast, [f]. Derefter trykkes på selve tasten.
- Valg af en funktion, der er trykt på selve tastens overflade, udføres ved at trykke direkte på tasten.
- Forvalg af en funktion, der er trykt med blåt på den skrå forside af en tast, udføres med den blå forvalgstast, [g]. Derefter trykkes på selve tasten.

Bemærk, at når der trykkes på [f] eller [g], vises **f** eller **g** i lyspanelet som tegn på, at den pågældende forvalgstast er aktiv. Indikatoren forsvinder først, når der trykkes på en funktionstast, der kan afslutte tastsekvensen.

Annullering af forvalgstaster

Hvis De kommer til at bruge en forkert forvalgstast, skal De trykke [f] CLEAR [PREFIX] Når [f] CLEAR [PREFIX] bruges efter [f], [g], [STO], [RCL], [GTO], [GSB], [f] [HYP] eller [g] [HYP^{-1}], bliver indtastningen annulleret. Da [f] [PREFIX] også bruges til udlæsning af den fulde mantisse af et tal, bliver lyspanelregistrets mantisse udlæst med 10 betydende cifre i ca. 1 sekund.

Funktioner af én variabel

Funktioner af én variabel virker direkte på tallet i lyspanelregistret – uden at påvirke noget andet registerindhold. HP-11C's funktioner af én variabel bruges således:

1. Indtast argumentet (hvis det da ikke allerede står i lyspanelregistret).
2. Tryk på den tilsvarende funktionstast (efter en eventuel forvalgstast).

Indtastning	**Lyspanel**
45	***45***
[g] [LOG]	***1,6532***

Funktioner af to variable

En funktion af to variable har tallet i lyspanelregistret som det ene argument og tallet i Y-registret (omtales senere) som det andet argument. De fire regningsarter er eksempler på funktioner af to variable.

[ENTER]. Hvis det ene af tallene, der skal bruges til en funktion af to variable, allerede står i lyspanelregistret, skal man ikke bruge [ENTER] til at skille første og anden operand. Skal man derimod indtaste begge, bruges [ENTER] til at afslutte den første operand.

En operation som f.eks. 2 ÷ 3 udføres således:

1. Indtast det første tal.

2. Tryk på [ENTER] for at skille det første tal fra det andet.
3. Indtast det andet tal.
4. Tryk på den tilsvarende funktionstast (i dette tilfælde [÷]).

Indtastning	Lyspanel
2	*2*
[ENTER]	*2,0000*
3	*3*
[÷]	*0,6667*

Indtastningsrækkefølge. I forbindelse med addition og multiplikation er operandernes orden ligegyldig, men ved subtraktion og division skal der skelnes nøje mellem minuend/dividend (1. operand) og subtrahend/divisor (2. operand). Det tal, der trækkes fra eller divideres med, skal altså indtastes som det andet tal.

Opgave	Indtastning	Lyspanel
10 – 3	10 [ENTER] 3 [–]	*7,0000*
3 – 10	3 [ENTER] 10 [–]	*-7,0000*
10 ÷ 3	10 [ENTER] 3 [÷]	*3,3333*
3 ÷ 10	3 [ENTER] 10 [÷]	*0,3000*

Når man arbejder med andre af HP-11C's funktioner af to variable, f.eks. [y^x] , skal det tal, der er betegnet x på funktionstasten, indtastes som det *sidste* tal. Hvis man f.eks. skal opløfte 2 til den 3. potens, løses opgaven ved at man først indtaster: 2 [ENTER] 3, hvorefter beregningen afsluttes med [y^x].

Indtastning	Lyspanel	
2 [ENTER]	*2,000*	
3	*3*	3 er x-værdien.
[y^x]	*8,0000*	2 *(y)* opløftet til 3. (x'te) potens.

Prøv dernæst at regne følgende opgaver. Bemærk, at [ENTER] kun skal bruges til at skille tal, der indtastes i umiddelbar rækkefølge. Et tidligere beregnet resultat (eller mellemresultat) vil automatisk give plads for et nyindtastet tal.

Eksempel: Bestem (2+4)÷8 og (9+17-4+24)÷4

Indtastning	**Lyspanel**	
2 [ENTER]	***2,000***	
4	***4***	
[+]	***6,000***	(2+4)
8	***8***	
[÷]	***0,7500***	(2+4)÷8
9 [ENTER]	***9,000***	
17 [+]	***26,0000***	(9+17)
4 [-]	***22,0000***	(9+17-4)
23 [+]	***45,0000***	(9+17-4+23)
4 [÷]	***11,2500***	(9+17-4+23)÷4

Selv endnu mere komplicerede beregninger udføres på samme enkle måde: automatisk lagring og fremkald af mellemresultater.

Eksempel: Bestem (6+7)×(9-3).
Først beregnes (6+7):

Indtastning	**Lyspanel**	
6	***6***	
[ENTER]	***6,0000***	
7	***7***	
[+]	***13,0000***	(6+7)

Dernæst beregnes (9-3). Da der nu skal indtastes et *nyt* talpar, skal [ENTER] bruges efter 9 for at skille 9-tallet fra det næste tal 3. Men da resultatet af (6+7) er et mellemresultat, der allerede står i lyspanelregistret, skal [ENTER] ikke bruges før 9. (9-3) bestemmes således:

Indtastning	**Lyspanel**	
9	***9***	
[ENTER]	***9,0000***	
3	***3***	
[-]	***6,0000***	(9-3)

Endelig ganges mellemresultaterne (13 og 6), så man får slutresultatet:

Indtastning	**Lyspanel**	
[×]	***78,0000***	(6+7)×(9–3) = 78

Bemærk, at HP-11C automatisk dels lagrer mellemresultater, dels henter dem frem, når der er brug for dem. Den finder selv ud af, i hvilken rækkefølge mellemresultaterne skal bruges som operander – den tager simpelthen de sidst lagrede mellemresultater først.
Uanset hvor kompliceret en beregning er, vil den altid kunne løses som en kombination af operationer med én eller to operander – ligesom De selv ville gøre det, hvis De skulle udføre beregningen med papir og blyant.

Husk altid:

- [ENTER] bruges til at adskille to tal, når disse indtastes lige efter hinanden.
- Et ciffer, der indtastes efter udførelsen af en beregning, betragtes som første ciffer i et nyt tal.
- Mellemresultater lagres og hentes automatisk således, at de sidst lagrede hentes først.

Nu skulle De selv kunne løse følgende opgaver med HP-11C. Indtast dem, som om de skulle løses manuelt med blyant og papir. Lad være med at tænke på mellemresultaterne – dem tager HP-11C sig af.

$(16 \times 38) - (13 \times 11) = 465{,}0000$
$(27 + 63) \div (33 \times 9) = 0{,}3030$
$(\sqrt{(16{,}38 \times 0{,}55)}) \div .05 = 60{,}0300$
$4 \times (17 - 12) \div (10 - 5) = 4{,}0000$

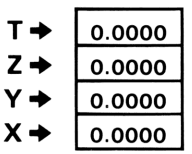
Afsnit 2

Den automatiske rullestak, Sidste X og dataregistre

Den automatiske rullestak og stakmanipulation

Det er den automatiske bevarelse og fremkald af mellemresultater, der gør HP-11C så let at bruge – og det er en automatisk rullestak, der sammen med [ENTER] gør det muligt at automatisere talbehandlingen. Rullestakken kan opfattes som en cylinder, der kan drejes om sin akse. Cylinderen er på langs delt op i fire lige store afsnit (registre). Af praktiske grunde afbilder man normalt rullestakken som fire kasser, der er stillet oven på hinanden. Kasserne (registrene) kaldes **X, Y, Z** og **T** – hvor X er nederst og T er øverst. Hvert register kan indeholde ét tal. Disse tal kaldes normalt *x*, *y*, *z* og *t*.

Den automatiske rullestak



Vises altid

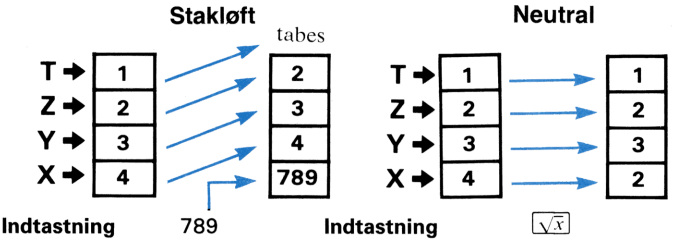
Når HP-11C er i normalstatus, vil tallet i lyspanelregistret være det samme som tallet i X-registret. I normalstatus er lyspanelet et vindue til den automatiske rullestak.

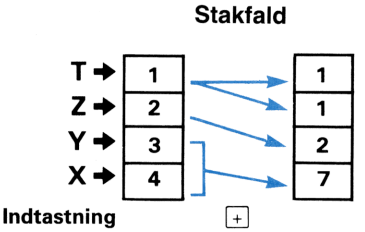
Ethvert tal, der indtastes og ethvert resultat, der beregnes, bliver lagret i X-registret og udlæst i lyspanelet.

Udførelsen af en funktion vil få tallene i rullestakken til at:

- skifte en plads opad (stakken løftes)
- blive hvor de er eller
- skifte en plads nedad (stakken falder).

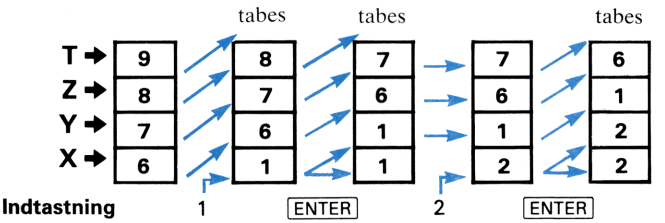
I de tre nedenstående eksempler starter stakken som vist til venstre, dernæst udføres den viste operation, hvorved stakken ændres som vist til højre.

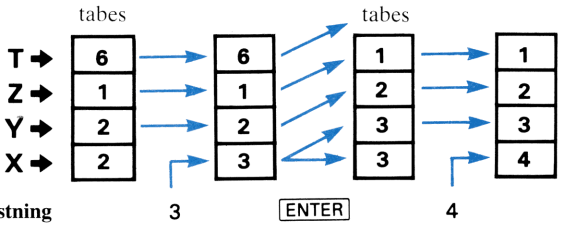




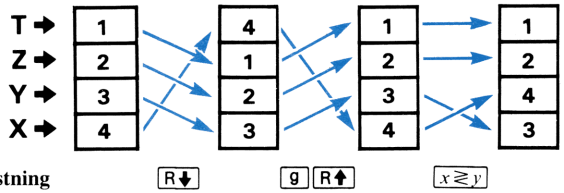
Stakmanipulation

ENTER. Bruges til at afslutte indtastning af et tal, når et andet tal følger umiddelbart efter. Når ENTER bruges, lægges en kopi af tallet i X-registret op i Y-registret. Man kan f.eks. *fylde* stakken med tallene 1, 2, 3 og 4 således:

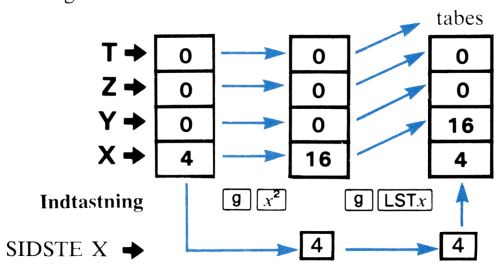




[R↓] *(rul-ned)*, [R↑] *(rul-op)* og [x≷y] *(X ombyttes med Y)*, [R↓] og [g] [R↑] bruges til at rotere stakken 90° eller ét register ad gangen uden tab af information. De kan f.eks. bruges til at gennemse stakken eller udvalgte registre. [x≷y] bytter om på indholdene af X-og Y-registrene. Hvis stakken oprindelig var fyldt med 1, 2, 3 og 4 vil [R↓], [g] [R↑] og [x≷y] medføre følgende ændringer:



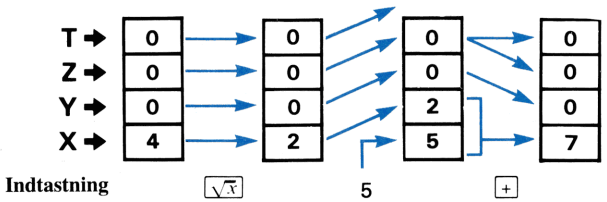
[LSTx] *(SIDSTE X)*. Når en funktion med et numerisk argument udføres, lægges en kopi af argumentet over i SIDSTE X, når funktionsværdien lagres i X. Trykker man [g] [LSTx] efter beregning af funktionen, vil argumentet blive lagret i X, medens funktionsværdien skubbes op i Y; [g] [LSTx] aktiverer stakløften. I Tillæg D angives for hver enkelt funktion, om den aktiverer stakløft, stakfald eller eventuelt er neutral, og om dens argument lagres i SIDSTE X – som f.eks. i nedenstående eksempel:



Funktioner og rullestakken

Når to tal skal indtastes efter hinanden, bruges [ENTER] til at adskille dem. Det er derimod ikke nødvendigt at bruge [ENTER], hvis tallet, der står i X-registret, er resultat af en eller anden operation – en egentlig funktion eller f.eks. [$x \gtrless y$].

Under talindtastning er X-registret i en særlig tilstand, som ændres, så snart der trykkes på en tast, der ikke vedrører talindtastning. De fleste funktioner aktiverer stakløften således, at indholdet af X-registret automatisk bliver skubbet op i Y-registret, når der indtastes et tal efter udførelsen af en funktion.



Nogle få funktioner: [ENTER], [g] [CLx], [$\Sigma+$] og [g] [$\Sigma-$] blokerer stakløften således, at det næste tal, der indtastes, vil overskrive indholdet af X-registret. Når [←] virker som [g] [CLx], dvs. nulstiller hele lyspanelregistret, blokerer den stakløften. I alle andre tilfælde er [←] neutral, dvs. helt uden betydning for stakløften.

Eksemplet, hvor stakken blev fyldt med 1, 2, 3 og 4, er en god illustration af [ENTER]'s blokering af stakløften.

Ovennævnte kan virke lidt indviklet, men er i virkeligheden blot en formalisering af nogle metoder, som falder så naturligt, at man næppe tænker på dem under brugen af maskinen.

Funktioner af to variable

Man må altid være meget omhyggelig med operandernes placering i rullestakken. Ved udførelsen af aritmetiske beregninger, skal tallene lagres i stakken i samme orden, som hvis beregningen skulle udføres med papir og blyant.

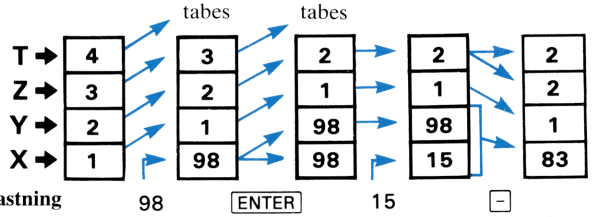
Hvis man f.eks. skal trække 15 fra 98, vil man først skrive »98« og så skrive »15« nedenunder – ligesom her:

$$\begin{array}{r} 98 \\ -15 \\ \hline \end{array}$$

Dernæst udføres subtraktionen

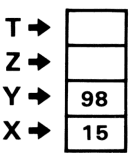
$$\begin{array}{r} 98 \\ -15 \\ \hline 83 \end{array}$$

Tallene lagres i rullestakken på præcis samme måde: det første tal, minuenden, lagres i Y og det andet tal, subtrahenden, lagres i X. Når der trykkes på [-], trækkes tallet i X-registret fra tallet i Y-registret. Herved falder stakken, så resultatet lagres i X-registret. Nedenstående viser, hvordan subtraktionen forløber:



Ligegyldigt hvilken aritmetisk operation der udføres, skal man først indtaste tallene, så de står i deres naturlige orden i rullestakken, derefter udføre operationen, hvorefter stakken falder. Man skulle bruge samme indtastning ved addition og multiplikation af 98 og 15 og ved division af 98 med 15.

$$\begin{array}{r} 98 \\ +15 \\ \hline \end{array} \qquad \begin{array}{r} 98 \\ \times 15 \\ \hline \end{array} \qquad \begin{array}{r} 98 \\ \hline 15 \end{array}$$

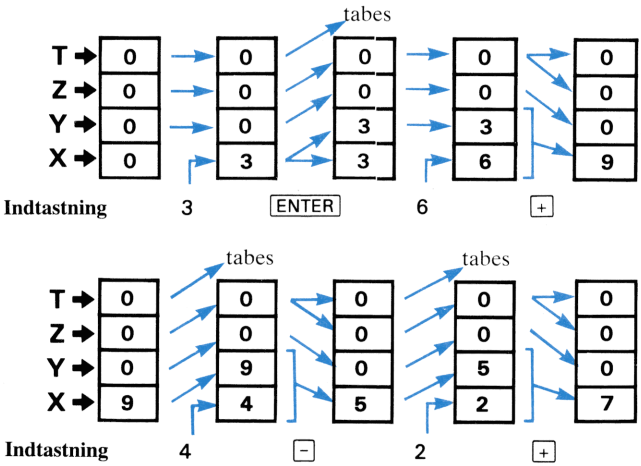


Kæderegning

Uanset om De mest vil bruge HP-11C til manuelle beregninger eller til programmeret problemløsning, er det sikkert, at De meget ofte vil få brug for kæderegning. Og det er først og fremmest på dette punkt, at HP-11C's indtastningssystem viser sin styrke. Selv under ekstremt komplicerede beregninger skal De kun udføre og holde styr på én beregning ad gangen. Den automatiske rullestak kan lagre op til 4 mellemresultater, indtil De får brug for dem. Hvad mere er, mellemresultaterne hentes automatisk frem. Processen er den samme som ved regning med papir og blyant. De har altid styr på beregningsgangen og kan bryde af, når som helst De ønsker det. Beregningen foregår på Deres betingelser.

De har allerede lært at indtaste et talpar med [ENTER] og derefter udføre en operation på de to tal. De har også set, hvordan stakken falder efter en funktion, og hvordan stakken løftes, når der igen indtastes tal.

For at illustrere disse stakmekanismer i forbindelse med kæderegning, vil vi betragte opgaven 3+6-4+2 = ? (stakken er nulstillet ved start).



Opgaven løses operation for operation, præcis som De selv ville gøre det. Stakken falder automatisk efter hver operation og løftes, når der indtastes et nyt tal. Selv langt mere indviklede opgaver løses på samme elegante måde.

Eksempel: I stedet for de pildiagrammer, der hidtil er brugt, vil vi bruge en tabelopsats til at anskueliggøre stakken og dens bevægelser ved beregning af

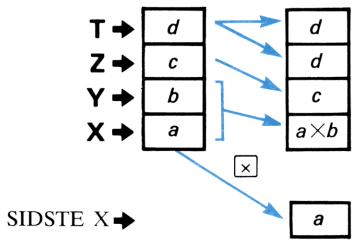
$$\frac{(3+4)\times(6-4)}{2}$$

T →	0	0	0	0	0	0
Z →	0	0	0	0	0	7
Y →	0	3	3	0	7	6
X →	3	3	4	7	6	6
Indtastning	3	ENTER	4	+	6	ENTER

T →	0	0	0	0	0
Z →	7	0	0	0	0
Y →	6	7	0	14	0
X →	4	2	14	2	7
Indtastning	4	−	×	2	÷

SIDSTE X

HP-11C's SIDSTE X-register er et særligt dataregister, der bruges til at lagre det tal, der stod i lyspanelregistret lige før udførelsen af en funktion.*



*) De statistiske funktioner, g x̄, g s og f L.R. er de eneste undtagelser. Disse funktioner beregner et resultat på basis af tal, der er lagret i dataregistre og tager således overhovedet ikke X-registrets indhold i betragtning.

Herved er der mulighed for at undgå genindtastning af tal, der skal bruges flere gange, og SIDSTE X er desuden til stor hjælp, hvis man begår fejl.

Eksempel: To forskellige tal, f.eks. 45,575 og 25,331 skal ganges med 0,175.

T →	0.0000	0.0000	0.0000	0.0000
Z →	0.0000	0.0000	0.0000	0.0000
Y →	0.0000	45.5750	45.5750	0.0000
X →	45.575	45.5750	0.175	7.9756

Indtastning → 45 [·] 575 [ENTER] [·] 175 [×]

SIDSTE X → [0.1750]

T →	0.0000	0.0000	0.0000
Z →	0.0000	7.9756	0.0000
Y →	7.9756	25.3310	7.9756
X →	25.331	0.1750	4.4329

Indtastning → 25.331 [g] [LST*x*] [×]

SIDSTE X → [0.1750] [0.1750] [0.1750]

[g] [LST*x*] er suveræn som hjælpemiddel efter fejl, hvor man f.eks. har brugt en forkert funktion eller indtastet et forkert tal og først opdager det efter at have udført operationen. Dividér 287 med 13,9 efter at der først er kludret i det med 12,9 som divisor.

Indtastning	**Lyspanel**	
287 [ENTER]	***287,0000***	
12,9 [÷]	***22,2481***	Forkert divisor!
[g] [LST*x*]	***12,9000***	Henter sidste x-værdi (divisoren) fra SIDSTE X.
[×]	***287,0000***	Udfører den *omvendte* operation – nu er dividenden i lyspanelet.
13,9 [÷]	***20,6475***	Udfører den rigtige operation.

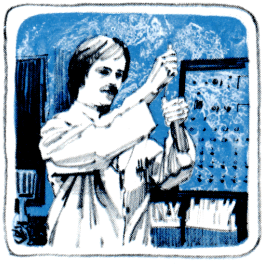
Konstantaritmetik

Tallet i T-registret bevares, når stakken falder. Det kan udnyttes til at holde et tal konstant ved aritmetiske operationer.

T ➡	*c*	*c*
Z ➡	*c*	*c*
Y ➡	*c*	*c*
X ➡	*x*	*cx*
Indtastning ➡		[×]

Når et tal skal bruges som konstant, skal man først fylde stakken med tallet ved at indtaste det og trykke tre gange på [ENTER].
Konstanten kan nu bruges: De skal blot indtaste det første argument og udføre det ønskede operationsforløb. Hver gang stakken falder, kopieres konstanten, så den hele tiden fylder stakken – bortset fra X-registret.

Eksempel: En bakteriolog arbejder med en bakteriestamme, hvis omfang typisk vokser med 15% om dagen. Hvis han starter med en kultur på 1000 bakterier, hvor stor en population har han da efter hver dag i 5 dage?



Metode: Brug [ENTER] 3 gange, så stakken fyldes op med vækstfaktoren (1,15). Indtast dernæst startpopulationen (1000). Efter hvert tryk på [×] udlæses populationen efter en ny dag. Brug [f] [FIX] 2 til at vælge samme udlæsningsformat som brugt i eksemplerne.

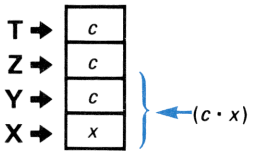
T ➡	0.00	0.00	0.00	1.15	1.15
Z ➡	0.00	0.00	1.15	1.15	1.15
Y ➡	0.00	1.15	1.15	1.15	1.15
X ➡	1.15	1.15	1.15	1.15	1,000

Indtastning 1.15 [ENTER] [ENTER] [ENTER] 1000

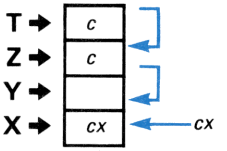
T →	1.15	1.15	1.15	1.15	1.15
Z →	1.15	1.15	1.15	1.15	1.15
Y →	1.15	1.15	1.15	1.15	1.15
X →	1,150.00	1,322.50	1,520.88	1,749.01	2,011.36
Indtastning	[×]	[×]	[×]	[×]	[×]

Første gang De trykker på [×], beregnes 1,15 × 1000. Resultatet (1.150,00) lagres i X-registret (og udlæses), stakken falder, og der dannes en ny kopi af konstanten i T-registret. Dvs., hver gang der trykkes på [×]:

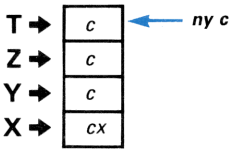
1. Udføres multiplikationen af tallene i X og Y.



2. Lagres resultatet af operationen i X, og resten af stakken falder.



3. Kopieres tallet i T samtidig med stakfaldet, så T holdes intakt.



Bemærk, at på grund af T-registrets selvkopiering, vil T, Z og Y hele tiden indeholde konstanten.

Denne metode kan kun bruges til kæderegning med en konstant. Hvis man skal udføre en række enkeltberegninger med samme tal, skal man slette det foregående resultat med [←], inden man indtaster den nye operand (husk, at [←] blokerer stakløften, hvis den kan bruges som [g] [CLx] – og det kan den jo netop efter en operation).

Man kan også bruge SIDSTE X, hvorved man undgår at skulle fylde stakken (som måske skal bruges til at lagre et par mellemresultater).

Bakteriologens problem ville kunne løses således:

Skift til fastkomma 4 med [f] [f] [FIX] 4, når De er færdig med opgaven.

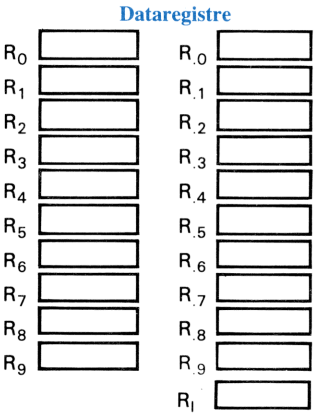
1. Indtast startpopulationen (1000) og tryk på [ENTER]
2. Indtast vækstfaktoren (1,15).
3. Tryk på [×], så populationen efter én dag udlæses.
4. Tryk [g] [LSTx] [×] og få udlæst populationen for næste dag.
5. Fortsæt med 4.

Brug af dataregistre

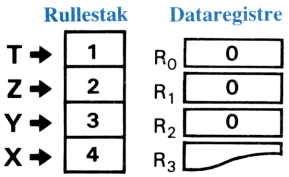
Lagring og fremkald af data er operationer, der involverer X-registret og HP-11C's 21 dataregistre, der er totalt uafhængige af rullestakken og SIDSTE X-registret.

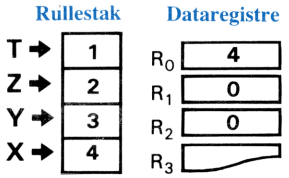
Lagring af tal

[STO] *(lagring)*. Når tasten efterfølges af en *adresse* (0 til 9, [·] 0 til [·] 9 eller [I]), kopieres tallet i lyspanelregistret over i det dataregister, der har den specificerede adresse.



Hvis ... ... og De trykker på [STO] 0, så...



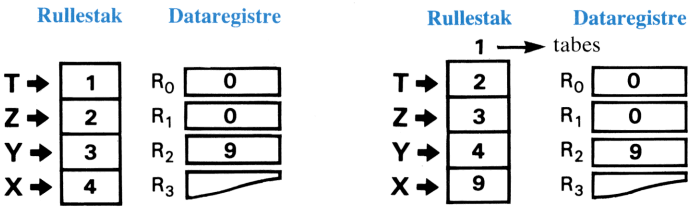


Det lagrede tal forbliver i dataregistret, indtil det overskrives af et andet tal, eller indtil alle dataregistre nulstilles, eller den kontinuerlige hukommelse slettes.

Fremkald af tal

RCL *(fremkald)*. Når tasten efterfølges af en *adresse* (0 til 9, · 0 til · 9 eller I), kopieres tallet i dataregistret med den specificerede adresse over i lyspanelregistret. Hvis stakløften ikke er blokeret, vil RCL løfte stakken.

Hvis ... ... og De trykker på RCL 2, så...



Eksempel på brug af dataregistre

Udfør følgende indtastninger:

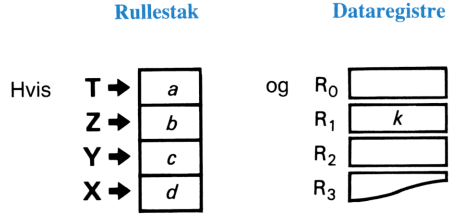
Indtastning	Lyspanel	
123	***123***	
STO 4	***123,0000***	Lagrer 123 i R_4.
678	***678***	
STO · 7	***678,0000***	Lagrer 678 i R_7.
RCL 4	***123,0000***	Fremkald af 123 fra R_4.
RCL · 7	***678,0000***	Fremkald af 678 fra R_7.

Nulstilling af dataregistre

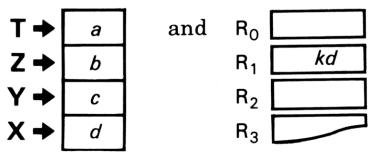
CLEAR REG (nulstilling af *dataregistre)*. Sætter indholdet af alle dataregistre til nul. **CLEAR** REG påvirker hverken rullestakken eller SIDSTE X-registret. Et enkelt dataregister nulstilles ved at lagre 0 i det.

Registeraritmetik

STO + , − , × , ÷ **n** *(registeraritmetik)*. Bruger tallet i X-registret og tallet i R_n til at udføre en aritmetisk operation med x som første operand, hvor resultatet lagres i R_n. Der kan direkte udføres registeraritmetik med dataregistrene R_0 til R_9. Med indirekte adressering (omtales i kapitel 2, afsnit 5) kan der desuden udføres registeraritmetik med registrene $R_{.0}$ til $R_{.9}$.



angiver status for rullestak og dataregistre, vil STO × 1 medføre:



Ingen ændring i rullestak

Tallet i R_1 ændres fra k til $(k \times d)$.

Eksempel på brug af registeraritmetik

Indtastning	Lyspanel	
18 STO 0	***18,0000***	Lagrer 18 i R_0.
3 STO ÷ 0	***3,0000***	Dividerer tallet i R_0 (18) med 3.
RCL 0	***6,0000***	Fremkald af tallet i R_0.
4 STO × 0	***4,0000***	Tallet i R_0 (6) ganges med 4.
RCL 0	***24,0000***	Fremkald af tallet i R_0.
STO + 0	***24,0000***	24 lægges til tallet i R_0.
RCL 0	***48,0000***	Fremkald af tallet i R_0.
40 STO − 0	***40,0000***	40 trækkes fra tallet i R_0.
RCL 0	***8,0000***	Fremkald af tallet i R_0.

Opgaver

1. Bestem værdien af x givet ved følgende udtryk:

$$x = \sqrt{\frac{8.33(4 - 5.2) \div ((8.33 - 7.46) \times 0.32)}{4.3(3.15 - 2.75) - (1.71 \times 2.01)}}$$

Svar: 4,5728

Løsningen kan f.eks. findes ved hjælp af:

4 ENTER 5.2 −
8.33 × g LSTx 7.46
− .32 × ÷
3.15 ENTER 2.75 −
4.3 × 1.71 ENTER
2.01 × − ÷ √x

2. Brug konstantaritmetik til at beregne saldoen for et lån på 1.000 kr., efter at der er betalt 6 fjortendages ydelser à 100 kr. Rentesatsen er 1% pr. termin (dvs. pr. fjortende dag).
   Metode: Fyld stakken med (1+i/100) (dvs. 1,01) og indtast den oprindelige sum – hovedstolen. Brug dernæst følgende formel til at beregne saldoen efter hver enkelt termin.

Ny saldo = ((Gammel saldo) × (1+i/100)) – Ydelse

Svar: 446,3186 (446,32)

3. 100 lagres i R_5, hvorefter følgende operationer udføres.
   1. Divider tallet i R_5 med 25.
   2. Træk 2 fra indholdet af R_5.
   3. Gang indholdet af R_5 med 0,75.
   4. Læg 1,75 til indholdet af R_5.
   5. Fremkald tallet i R_5.

Svar: 3,2500.

Afsnit 3

Matematiske funktioner

HP-11C's funktionssæt giver Dem adgang til at udføre en lang række operationer, der involverer ændring af et udlæst tal, matematiske standardfunktioner og statistik. Alle funktioner bruges på samme måde, uanset om de anvendes manuelt eller under programkontrol.

Pi

[f] [π] lagrer en 10-cifret tilnærmelse af π (3,141592654) i lyspanelregistret. Hvis rullestakken ikke er blokeret, vil [f] [π] løfte stakken.

Ændring af udlæst tal

Fortegnsskift. Ud over [CHS] *(fortegnsvender)* har HP-11C fire funktioner, der kan bruges til at ændre formatet af det udlæste tal: [g] [RND], [g] [ABS], [g] [INT] og [f] [FRAC].

Afrunding. Når der trykkes på [g] [RND], vil tallet i lyspanel*registret* blive afrundet, så det nøje svarer til det tal, der faktisk udlæses. Dvs. at tallet afrundes til det antal cifre/decimaler, der er specificeret i forbindelse med [f] [FIX], [f] [SCI] eller [f] [ENG].

Numerisk værdi. Når der trykkes på [g] [ABS], erstattes tallet i lyspanelregistret med tallets numeriske værdi.

Heltalsdel. [g] [INT] erstatter tallet i lyspanelregistret med tallets heltalsdel. Det betyder, at alle decimaler sættes til 0.

Decimaldel. [f] [FRAC] erstatter tallet i lyspanelregistret med tallets decimaldel. Det betyder, at tallets heltalsdel skæres væk.

Beregning af	Eksempel	Lyspanel
Afrunding	1,23456789	***1,23456789***
(ved [FIX] 4)	[g] [RND]	***1,2346***
Se hele mantissen	[f] [PREFIX]	***1,234600000***

Beregning af	Eksempel	Lyspanel
Numerisk værdi	12345 [CHS]	*-12345*
	[g] [ABS]	*12345,0000*
Heltalsdel	123,4567	*123,4567*
	[g] [INT]	*123,0000*
Decimaldel	123,4567	*123,4567*
	[f] [FRAC]	*0,4567*

Funktioner af én variabel

HP-11C's funktioner af én variabel har følgende karakteristika:

- Tallet i X-registret (lyspanelregistret) bruges som argument.
- Funktionsværdien lagres i X-registret, og argumentet overføres til SIDSTE X.
- Tallene i Y- Z- og T-registrene påvirkes ikke.

Generelle funktioner

Reciprokværdi. [1/x] beregner og udlæser den reciprokke værdi af tallet i X-registret, dvs. 1 divideret med tallet.

Fakultet og Gammafunktionen. [f] [x!] beregner og udlæser fakultet eller gammafunktionen af tallet i X-registret:

1. Fakultet
Når tallet i X-registret er et positivt heltal, n, mindre end 70, vil [f] [x!] beregne n fakultet, n! = 1 × 2 × 3 × ... × n.

2. Gammafunktionen
[f] [x!] kan også bruges til beregning af gammafunktionen, $\Gamma(x)$, der indgår i en lang række matematiske sammenhænge – specielt i forbindelse med statistikkens fordelingsfunktioner*. [f] [x!] beregner $\Gamma\ (\chi+1)$; skal gammafunktionen beregnes for et givet tal, må man således trække 1 fra tallet, inden man trykker på [f] [x!].

*) Gammafunktionen er en generaliseret fakultetsfunktion, der kan have alle reelle tal som argument. Når x er et positivt heltal, n, har man: $\Gamma(x+1) = \Gamma(n+1) = n!$

Kvadratrod. $\boxed{\sqrt{x}}$ beregner og udlæser kvadratroden af tallet i X-registret.

Kvadrering. $\boxed{g}$ $\boxed{x^2}$ kvadrerer og udlæser tallet i X-registret.

Beregning af	Eksempel	Lyspanel
Reciprokværdi	25	*25*
	$\boxed{1/x}$	*0,0400*
Fakultet	8	*8*
	$\boxed{f}$ $\boxed{x!}$	*40.320,0000*
Gammafunktionen	2,7	*2,7*
	$\boxed{\text{ENTER}}$ 1 $\boxed{-}$	*1,7000*
	$\boxed{f}$ $\boxed{x!}$	*1,5447*
Kvadratrod	3,9	*3,9*
	$\boxed{\sqrt{x}}$	*1,9748*
Kvadrering	12,3	*12,3*
	$\boxed{g}$ $\boxed{x^2}$	*151,2900*

Trigonometriske funktioner

De seks grundlæggende trigonometriske funktioner tager argumenter i det vinkelmål, der er specificeret, og som markeres i lyspanelet.

1. **Vinkelmål.** Skift af vinkelmål medfører ikke konvertering fra det oprindelige vinkelmål til det nye. Valg af vinkelmål indebærer udelukkende en angivelse af, hvordan HP-11C skal fortolke trigonometriske funktioners argumenter.

Grader. Vælges ved hjælp af $\boxed{g}$ $\boxed{\text{DEG}}$. I lyspanelet vises gradindikatoren DEG ikke.

Radian. Vælges ved hjælp af $\boxed{g}$ $\boxed{\text{RAD}}$. I lyspanelet vises radianindikatoren RAD.

$\boxed{g}$ $\boxed{\text{RAD}}$ **0.0000** RAD

Nygrader. Vælges ved hjælp af $\boxed{g}$ $\boxed{\text{GRD}}$. I lyspanelet vises nygradindikatoren GRAD (engelsk Grades).

$\boxed{g}$ $\boxed{\text{GRD}}$ **0.0000** GRAD

Der vil altid være valgt ét af disse vinkelmål. Den kontinuerlige hukommelse bevarer det sidst valgte vinkelmål, når der slukkes for maskinen. Efter en længere strømafbrydelse eller efter nulstilling af hele hukommelsen (se side 19), bliver DEG automatisk valgt som vinkelmål.

2. Trigonometriske funktioner

FIX

Tryk på	Beregner
SIN	sinus (sin)
g SIN^{-1}	Arcus sinus (Arcsin)
COS	cosinus (cos)
g COS^{-1}	Arcus cosinus (Arccos)
TAN	tangens (tg)
g TAN^{-1}	Arcus tangens (Arctg)

Inden udførelsen af disse funktioner må man altid sikre sig, at det ønskede vinkelmål er valgt. Bemærk, at for de inverse funktioner beregnes hovedværdierne (de principale værdier).

Beregning af	Eksempel	Lyspanel
(Grader er valgt som vinkelmål)		
Alle trigonometriske funktioner, f.eks.		
sinus	33,5	***33,5***
	SIN	***0,5519***
Arcus sinus	,7982	***0,7982***
	g SIN^{-1}	***52,9586***

Omregning af tid og vinkler

HP-11C kan fortolke tal, der repræsenterer tid eller vinkler, på to måder: som decimaltal eller som timer (grader/minutter/sekunder, og der kan omregnes mellem disse formater. HP-11C benævner timer med »H« – efter det engelske Hours (h er SI-standardens betegnelse for timer).

Timer.Decimaltimer (H.h)	til Timer.Minutter/Sekunder/Decimalsekunder (H.MMSSs)
Grader.Decimalgrader (D.d)	til Grader.Minutter/Sekunder/Decimalsekunder (D.MMSSs)

Omregning fra timer (grader) til timer (grader)/minutter/sekunder.

H.h –> H.MMSSs
eller
D.d –> D.MMSSs

f →H.MS omregner tallet i X-registret fra decimaltimer(grader) til timer(grader)/min/sek.

Omregning fra timer(grader)/minutter/sekunder til timer(grader)

H.MMSSs –> H.h
eller
D.MMSSs –> D.d

g →H omregner tallet i X-registret fra timer(grader)/min/sek til timer.

Omregning mellem grader og radian

g →DEG og f →RAD bruges til at omregne tallet i lyspanelet fra og til *decimal*grader og radian. D.d ➡ R.r og R.r ➡ D.d

Omregning fra grader til radian. f →RAD omregner tallet i X-registret fra grader til radian.

Omregning fra radian til grader. g →DEG omregner tallet i X-registret fra radian til grader.

Omregning fra-til	Eksempel	Lyspanel
Decimaltimer (H.h) eller grader (D.d) til H.MMSSs eller D.MMSSs	17,553 f →H.MS	***17,553*** ***17,3311***
Decimalerne kan udlæses i FIX 4 med:	f PREFIX	***1733108000***
H.MMSSs eller D.MMSSs til decimaltimer (H.h) eller grader (D.d).	12.3045 g →H	***12,3045*** ***12,5125***
Grader til radian.	40,5 f →RAD	***40,5*** ***0,7069***
Radian til grader.	1,1746 g →DEG	***1,1746*** ***67,2996***

Logaritmefunktioner

Naturlig logaritme. [g] [LN] beregner og udlæser den naturlige logaritme af tallet i X-registret. Den naturlige logaritme er logaritmefunktionen med e (2,718281828) som grundtal.

Eksponentialfunktionen. [e^x] beregner og udlæser eksponentialfunktionen for ×. Dvs. opløfter e (2,718281828) til den ×'te potens (hvor × er tallet i X-registret). Eksponentialfunktionen er antilogaritme til den naturlige logaritme.

10-talslogaritmen. [g] [LOG] beregner og udlæser 10-talslogaritmen af tallet i X-registret. 10-talslogaritmen er logaritmefunktionen med 10 som grundtal. Kaldes også den briggske logaritme eller almindelig logaritme.

Almindelig antilogaritme. [10^x] beregner og udlæser den almindelige antilogaritme af tallet i X-rgistret – dvs. opløfter 10 til den ×'te potens.

Beregning af	Eksempel	Lyspanel
Naturlig logaritme	45	***45***
	[g] [LN]	***3,8067***
Eksponentialfunktionen	3,4012	***3,4012***
	[e^x]	***30,0001***
10-talslogaritme	12,4578	***12,4578***
	[g] [LOG]	***1,0954***
Almindelig antilogaritme	3,1354	***3,1354***
	[10^x]	***1.365,8405***

Hyperbolske funktioner

Tryk på	Beregner
[f] [HYP] [SIN]	Hyperbolsk sinus *(sinh)*
[g] [HYP^{-1}] [SIN]	Invers hyperbolsk sinus *(arcsinh)*
[f] [HYP] [COS]	Hyperbolsk cosinus *(cosh)*
[g] [HYP^{-1}] [COS]	Invers hyperbolsk cosinus *(arccosh)*
[f] [HYP] [TAN]	Hyperbolsk tangens *(tgh)*
[g] [HYP^{-1}] [TAN]	Invers hyperbolsk tangens *arctgh)*

Beregning af	Eksempel	Lyspanel
Alle hyperbolske funktioner, f.eks. hyperbolsk sinus	2,53 f HYP SIN	*2,53* *6,2369*
Invers hyperbolsk sinus	1,95 g HYP^{-1} SIN	*1,95* *1,4210*

Funktioner af to variable

HP-11C beregner alle funktioner af to variable ved at tage tallene i X- og Y-registrene som argumenter. Når argumenterne skal indtastes, skal man altid indtaste Y-værdien først, trykke på ENTER, indtaste X-værdien og derefter trykke på funktionstasten.

Potensopløftning. f y^x beregner og udlæser værdien af y opløftet til x'te potens, hvor y er tallet i Y-registret, og x er tallet i X-registret.

Beregning af	Eksempel	Lyspanel
Potens	2 ENTER 3 f y^x	*2,000* *3* *8,0000*

Procent

Man finder en bestemt procent af et tal ved at:

1. Indtaste grundtallet.
2. Trykke på ENTER.
3. Indtaste procentsatsen.
4. Trykke på g %.

T →				
Z →				
Y →		150	150	150
X →	150	150	25	37.5
Indtastning	150	ENTER	25	g %
SIDSTE X				25

Procenten lagres i X-rgistret, grundtallet bevares i Y-registret, og procentsatsen overføres til SIDSTE X. Det betyder, at den beregnede procent uden videre kan lægges til, trækkes fra osv. Stakken løftes ikke af [g] [%], så tal i Z- og T-registrene er intakte.
Ovenstående eksempel viser hvordan man beregner 25% af 150.

Procentisk forskel

[g] [Δ%] beregner og udlæser den procentiske forskel – den relative forskel – mellem tallene i X- og Y-registrene.
Man finder den procentiske forskel mellem to tal ved at:

1. Indtaste grundtallet (typisk det tidsmæssigt først forekommende).
2. Trykke på [ENTER].
3. Indtaste det andet tal.
4. Trykke på [g] [Δ%].

T→				
Z→				
Y→		150	150	150
X→	150	150	225	50
	150	[ENTER]	225	[g] [Δ%]
SIDSTE X				225

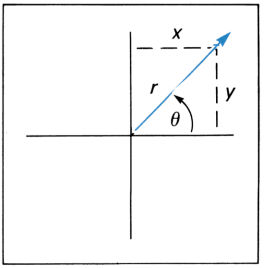
Bruges ovenstående indtastningsform, vil en absolut stigning give et positivt resultat, hvorimod et absolut fald vil give et negativt resultat.

Ovenstående eksempel viser, hvordan man beregner den procentiske forskel mellem 150 (grundtal) og 225.

Beregning af	Eksempel	Lyspanel
Procent		
Grundtal	200 [ENTER]	***200,0000***
Procentsats	75	***75***
Procent	[g] [%]	***150,0000***
Procentisk forskel		
Grundtal	40 [ENTER]	***40,0000***
Andet tal	160	***160***
Procentisk forskel	[g] [Δ%]	***300,0000***

Omregning mellem polære og rektangulære koordinater

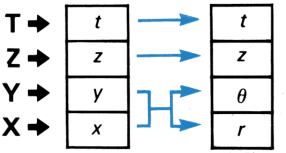
HP-11C har to funktioner, f →R og g →P, til omregning mellem polære og rektangulære koordinater.



Vinklen θ regnes for at være i det vinkelmål, der er valgt til trigonometriske beregninger. θ måles som vist på tegningen til højre, og θ vil som resultat altid ligge mellem -180° og 180°.

Omregning til polære koordinater

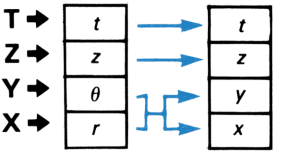
g →P lader tallene i X- og Y- registrene repræsentere rektangulære koordinater (x, y), som omregnes til polære koordinater (r, θ).



Indtastning g →P

Omregning til rektangulære koordinater

f →R lader tallene i X- og Y- registrene repræsentere polære koordinater (r, θ), som omregnes til rektangulære koordinater (x, y).



Indtastning f →R

Omregning	Eksempel	Lyspanel
Rektangulære til polære koordinater:		
y	5 ENTER	*5,0000*
x	10	*10*
r	g →P	*11,1803*
θ	$x \gtrless y$	*26,5651*
Polære til rektangulære koordinater:		
θ	30 ENTER	*30,0000*
r	12	*12*
x	f →R	*10,3923*
y	$x \gtrless y$	*6,0000*

Sandsynlighedsregning

Permutationer. [f] [$P_{y,x}$] beregner og udlæser antallet af mulige permutationer af y forskellige genstande taget x ad gangen, idet samme genstande i forskellig indbyrdes orden tælles med (f.eks. medregnes både ABC og ACB ved (P 6,3) for ABCDEF, og ingen genstand medtages mere end én gang i en enkelt permutation). [f] [$P_{y,x}$] beregner antallet af permutationer ved hjælp af:

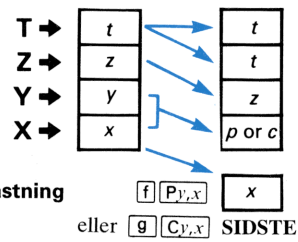
$$P_{y,x} = \frac{y!}{(y-x)!}$$

Kombinationer. [g] [$C_{y,x}$] beregner og udlæser antallet af mulige kombinationer af y forskellige genstande taget x ad gangen, idet der ikke tages hensyn til genstandenes indbyrdes orden (f.eks. medregnes ikke både ABC og ACB ved (C 6,3) for ABCDEF, og ingen genstand medregnes mere end én gang i en enkelt kombination).
[g] [$C_{y,x}$] beregner antallet af kombinationer ved hjælp af:

$$C_{y,x} = \frac{y!}{x!(y-x)!}$$

For to givne tal beregnes antallet af permutationer eller kombinationer således:

1. Indtast antallet af forskellige genstande *(y)*.
2. Tryk på [ENTER].
3. Indtast antallet af genstande i hver permutation/kombination *(x)*.
4. Tryk på [f] [$P_{y,x}$] eller [g] [$C_{y,x}$].



Resultatet, p eller c, udlæses i X-registret, stakken falder, og antallet af genstande pr. permutation/kombination *(x)* lagres i SIDSTE X.

Alle tal, der bruges ved beregning af permutationer og kombinationer, skal være positive heltal.

Bemærk: Beregningstiden for [$P_{y,x}$] og [$C_{y,x}$] kan være flere sekunder eller endog endnu længere afhængigt af størrelsen af *y, x* og slutresultatet. De største tal, der kan bruges ved beregning af permutationer og kombinationer, er 10^{10}-1.
Under beregning af permutationer eller kombinationer vises ***running*** i lyspanelet.

Beregning af	Eksempel	Lyspanel
Permutationer:		
Eksempel: 10 genstande		
taget 3 ad gangen.	10 [ENTER]	***10,0000***
	3 [f] [$P_{y,x}$]	***720,0000***
Kombinationer:		
Eksempel: 10 genstande		
taget 3 ad gangen.	10 [ENTER]	***10,0000***
	3 [g] [$C_{y,x}$]	***120,0000***

Statistik

Pseudotilfældige tal

HP-11C kan generere jævnt fordelte, pseudotilfældige tal*. Før hver sekvens skal der opgives et udgangstal, hvor man enten kan bruge maskinens standard (som er 0) eller selv indtaste en værdi *u*, hvor $0 < u < 1$. Udgangstallet bevares af den kontinuerlige hukommelse.

Udgangstal. Sekvensen af pseudotilfældige tal kan til enhver tid initialiseres ved at lagre et nyt udgangstal. Man indtaster et tal mellem 0 og 1 og trykker [STO] [f] [RAN#]. Et pseudotilfældigt tal generes ved at trykke [f] [RAN#] hvorefter der udlæses et tal.
[f] [RAN#] påvirker rullestakken på samme måde som fremkald af et tal fra et dataregister. Det senest genererede pseudotilfældige tal bliver udgangstal for det næste. Det betyder bl.a., at man kan repetere en sekvens, hvis blot man kender udgangstallet for det første tal i sekvensen.

*) **Metoden består den spektraltest, som er beskrevet i** *Donald E. Knuth:* The Art of Computer Programming, Vol. 2, Afsnit 3.3.4.

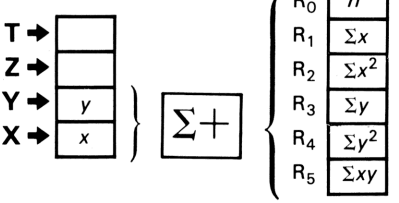
Generator for pseudotilfældige tal	Eksempel	Lyspanel
Lagring af f.eks. 0,5674 som udgangstal		
(Udgangstal)	[.] 5764	*0,5764*
Lagring af udgangstal	[STO]	
	[f] [RAN#]	*0,5764*
Generering af en sekvens af pseudotilfældige tal på basis af det lagrede udgangstal	[f] [RAN#]	*0,3422*
	[f] [RAN#]	*0,2809*

Hvis man bruger et tal, som er større end 1, vil udgangstallet automatisk blive divideret med en så passende potens af 10, at tallet lagres som et tal mellem 0 og 1. F.eks. vil 123,0 og 12,3 begge blive lagret som 0,123.

I kapitel 3, afsnit 2 diskuteres brugen af tilfældige tal.

Summation af statistikdata

[Σ+] er en funktion af to variable, som bruges til at danne forskellige summer af tal, der indtastes i X- og Y-registrene. Summerne lagres automatisk i registrene R_0 til R_5.

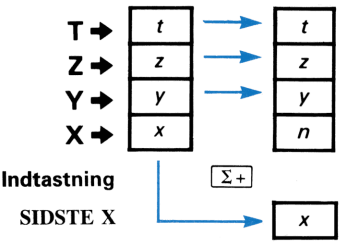


Man bør altid nulstille registrene R_0 til R_5, inden man begynder at summere et nyt datasæt. HP-11C har en funktion, der gør det muligt at nulstille alle sumregistrene på én gang: [f] CLEAR [Σ]. Det skal bemærkes, at også rullestakken nulstilles ved denne operation.

Når der trykkes på [Σ+], summeres data efter følgende skema:

Register	Indhold
R_0	n Antallet af datasæt. (n udlæses desuden i lyspanelet).
R_1	Σx Sum af x-værdierne.
R_2	Σx^2 Kvadratsum af x-værdierne.
R_3	Σy Sum af y-værdierne.
R_4	Σy^2 Kvadratsum af y-værdierne.
R_5	Σxy Produktsum af x- og y-værdierne.

Når [Σ+] bruges, bliver x-værdien overført til SIDSTE X, og den nye værdi for n udlæses. *Y-registrets indhold ændres ikke.*



[Σ+] blokerer stakløften således, at n overskrives af det næste tal, der indtastes.

Hvis der kun skal arbejdes med én variabel (x) i stedet for to (x og y), skal man sikre sig, at Y-registret er nulstillet før hver brug af [Σ+]. Bruges [f] **CLEAR** [Σ] umiddelbart før summationen, nulstilles hele stakken og således også Y-registret. Hvis man udelukkende summerer og ikke foretager aritmetiske operationer, vil Y-registret forblive nulstillet.

Nogle datasæt består af en række x- (og/eller y-værdier), der kun afviger ganske lidt fra hinanden. I sådanne tilfælde kan man få udført beregningerne langt mere nøjagtigt, hvis man, for hvert enkelt tal, summerer differensen mellem tallet og et pænt, rundt tal i nærheden af middelværdien. Dette tal skal så senere lægges til de beregnede værdier for $\bar{x}$, $\hat{y}$, skæringspunkt på Y-akse osv.

Hvis f.eks. x-værdierne er 665999, 666000 og 666001, bør man lagre 666000 i f.eks. R_6 og normalisere datasættet ved at trække dette tal fra de enkelte x-værdier; man summerer altså -1, 0 og 1. Efter beregning af $\bar{x}$ fås det rigtige resultat ved at lægge 666000 til det foreløbige resultat.

Begrundelsen for dette krumspring er, at der tabes betydende cifre ved arbejde med meget store tal – så i de tilfælde, hvor en del af cifrene ikke bidrager med nogen unik information, bør man skaffe sig af med disse redundante cifre.

I nogle tilfælde kan HP-11C ikke beregne s, r, L.R. og $\hat{y}$, når summerne ligger for tæt på hinanden. Hvis man forsøger at udføre en sådan beregning, vil maskinen give fejlmeddelelsen ***Error 2.*** Dette vil ikke ske, hvis man normaliserer sine data som omtalt ovenfor.

> **Bemærk:** I forbindelse med almindelig registeraritmetik markeres overløb i et dataregister med ***Error 1.*** Men ved summation med [Σ+] vil overløb i R_0 til R_5 ikke give fejlmeddelelse.

Data i summationsregistrene kan fremkaldes med [RCL] efterfulgt af nummeret på det register, der indeholder det ønskede resultat. Hvis man vil have fremkaldt både Σx og Σy, kan dette gøres samtidigt ved at trykke på [RCL] [Σ+]. Herved kopieres Σx fra R_1 over i X-registret, medens Σy kopieres fra R_2 over i Y-registret. Stakken løftes på præcis samme måde, som hvis De havde fremkaldt de to summer på sædvanlig vis.

Eksempel: Energiforskeren Rigmor I. Voltz aner en sammenhæng mellem stigningen i verdens samlede kulproduktion i årene 1972-1976 og det globale elforbrug i samme periode. Voltz vil bruge HP-11C til at analysere de givne data og finder til den ende Σx, Σx^2, Σy, Σy^2 samt Σxy for (x, y)-værdierne i materialet.

År	1972	1973	1974	1975	1976
Kulproduktion *(y)* (milliarder tons)	1,761	1,775	1,792	1,884	1,943
Elforbrug *(x)* (milliarder MWh)	5,552	5,963	6,135	6,313	6,713

Indtastning	Lyspanel	
[f] CLEAR [Σ]	***0,0000***	Nulstilling af summationsregistrene og rullestakken.
1,761 [ENTER]	***1,761***	1972-data.
5,552 [Σ+]	***1,0000***	

Indtastning	Lyspanel	
1,775 [ENTER]	***1,775***	
5,963 [Σ+]	***2,0000***	1973-data.
1,792 [ENTER]	***1,792***	
6,135 [Σ+]	***3,0000***	1974-data.
1,884 [ENTER]	***1,884***	
6,313 [Σ+]	***4,0000***	1975-data.
1,943 [ENTER]	***1,943***	
6,713 [Σ+]	***5,0000***	1976-data.
1 [RCL]	***30,6760***	Sum af x-værdier (Σx) hentes fra R_1.
2 [RCL]	***188,9386***	Kvadratsum for x-værdier (Σx^2) hentes fra R_2.
3 [RCL]	***9,1150***	Sum af y-værdier (Σy) hentes fra R_3.
4 [RCL]	***16,7877***	Kvadratsum for y-værdier (Σy^2) hentes fra R_4.
5 [RCL]	**56,2924**	Produktsummen for x- og y-værdier (Σxy) hentes fra R_5.

Korrektion af summerede data

Hvis De opdager, at De er kommet til at indtaste et forkert datapar, kan fejlen let rettes.

Metoden bruges også, selvom kun det ene af tallene er forkert.

1. Indtast det *forkerte* datapar, så det placeres som normalt i X- og Y-registrene.
2. Tryk på [g] [Σ−], hvorved summationsregistrenes indhold korrigeres.
3. Indtast det *rigtige* datapar, *(x, y)*.
4. Tryk på [Σ+].

Hvis De opdager fejlen umiddelbart efter tryk på [Σ+], kan fejlen korrigeres ved at hente x i SIDSTE X. Derved står x i X og y i Y, så [g] [Σ−] kan bruges direkte.

Bemærk: Selvom [g] [Σ−] kan bruges til at korrigere et fejlindtastet datapar *(x, y)*, kan funktionen ikke ophæve eventuelle ophobede afrundingsfejl. Man kan i princippet komme ud for at summere en række data med [Σ+], slette dem alle med [g] [Σ−] og alligevel ikke have nulstillede summationsregistre.

Fejlen vil normalt være ganske lille og ubetydelig.
Der er dog en enkelt type fejl, hvor det bedst vil kunne svare sig at starte helt forfra: hvis det forkerte tal er størrelsesordener større end det rigtige tal, vil det gå ud over antallet af betydende cifre.

Eksempel. Efter indtastning af de foregående data finder Voltz andre kilder, der får hende til at tro på, at kulproduktionen i 1976 snarere var 1,946 mia tons end 1,943 mia tons. Med [g] [Σ-] fjernes de oprindelige 1976-data fra summerne, og de korrigerede data indlæses med [Σ+].

Indtastning	**Lyspanel**	
1,943 [ENTER]	***1,9430***	Indtastning af det datapar,
6,713 [g] [Σ-]	***4,0000***	der skal trækkes ud af summerne. I lyspanelet vises, at summerne nu er baseret på 4 datapar.
1,946 [ENTER]	***1,9460***	Indtastning og summering af
6,713 [Σ+]	***5,0000***	de korrigerede data. Nu er der igen summeret 5 datapar.

Gem de foregående summer i Deres HP-11C – de skal bruges i de følgende eksempler.

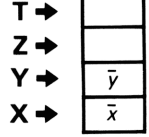
Middelværdi

[g] [$\bar{x}$] beregner den aritmetiske middelværdi af de indtastede og summerede x- og y-værdier.
Når der trykkes på [g] [$\bar{x}$], sker følgende:

1. Stakken løftes på samme måde som beskrevet for [RCL] [Σ+].
2. Middelværdien af x-værdierne ($\bar{x}$) beregnes ved hjælp af tallene, der er lagret i R_1 (Σx) og R_0 *(n)*. Middelværdien af y-værdierne ($\bar{y}$) beregnes tilsvarende ved hjælp af tallene i R_3 (Σy) og R_0 *(n)*.

$$\bar{x} = \frac{\Sigma x}{n} \qquad \bar{y} = \frac{\Sigma y}{n}$$

[g] [$\bar{x}$] =

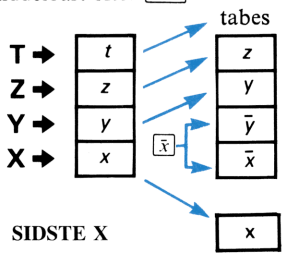


$\bar{x}$ og $\bar{y}$ lagres i henholdsvis X- og Y-registret.

Eksempel. Beregn middelværdierne for verdens kulproduktion og elforbrug i årene 1972-76 ved hjælp af de indtastede og korrigerede data.

Indtastning	Lyspanel	
[g] [$\bar{x}$]	***6,1352***	Gennemsnitligt elforbrug i den pågældende femårs periode.
[$x \gtrless y$]	***1,8316***	Gennemsnitlig kulproduktion i samme periode.

Nedenstående figur viser, hvad der sker med tallene i rullestakken, når tasten bruges umiddelbart efter [$\Sigma+$]



Bevar stadig tallene i summationsregistrene, så de kan bruges i næste eksempel.

Standardafvigelse

[g] [s] beregner standardafvigelsen af de summerede data. Standardafvigelsen, eller spredningen, er mål for de enkelte værdiers afvigelse fra middelværdien. HP-11C beregner standardafvigelserne af x-værdierne (s_x) og af y-værdierne (s_y).

$$s_x = \sqrt{\frac{n\Sigma x^2 - (\Sigma x)^2}{n(n-1)}} \qquad s_y = \sqrt{\frac{n\Sigma y^2 - (\Sigma y)^2}{n(n-1)}}$$

Disse formler giver de *bedste estimater* for standardafvigelserne af populationer på basis af udtagne stikprøver. Det er derfor almindelig praksis at referere til denne type spredninger som *stikprøve*-standardafvigelser.

Når der trykkes på [g] [s], sker følgende:

1. Stakken løftes på samme måde som beskrevet for [RCL] [Σ+].
2. Standardafvigelsen af x-værdierne (s_x) beregnes ved hjælp af tallene, der er lagret i R_2 (Σx^2), R_1 (Σx) og R_0 (n). s_x lagres i X-registret.
3. Standardafvigelsen af y-værdierne (s_y) beregnes ved hjælp af tallene i R_4 (Σy^2), R_3 (Σy) og R_0 (n). s_y lagres i Y-registret.

Eksempel. Beregn standardafvigelserne af den korrigerede kulproduktion og elforbruget, idet tallene fra de foregående eksempler stadig benyttes.

Indtastning	**Lyspanel**	
[g] [s]	***0,4287***	Standardafvigelse af elforbruget (s_x) for fem-års perioden.
[$x \gtrless y$]	***0,0800***	Standardafvigelsen af kulproduktionen (s_y) for fem-års perioden.

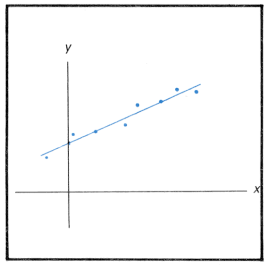
Der arbejdes videre med dette eksempel, så De skal ikke slette tallene i registrene R_0 til R_5.

Rullestakken påvirkes af [g] [s] på præcis samme måde, som vi før så det for [g] [$\bar{x}$].

Hvis data udgør hele populationen og ikke blot en stikprøve, skal man bruge *populations*-standardafvigelsen, σ. σ er givet ved $s\sqrt{(n-1)\div n}$ og kaldes også standardafvigelse med N-vægtning (hvor stikprøve-standardafvigelsen så har (N-1)-vægtning). Forskellen på σ og s er normalt meget lille, medmindre n er meget lille. Men hvis man ønsker at bestemme σ, gøres det lettest ved at lægge middelværdierne til datasættet og derefter bestemme standardafvigelserne, [g] [$\bar{x}$] [Σ+] [g] [s].

Lineær regression

Lineær regression er en statistisk metode, som bruges til at finde den rette linie, der, ifølge visse kriterier, passer bedst til et sæt på mindst to-dimensionale datapunkter. Når man har summeret datapunkterne med [Σ+], bruges [f] [L.R.] til at bestemme koefficienterne i den lineære ligning $y = Ax + B$ ved hjælp af *mindste kvadraters metode*.



Når [f] [L.R.] bruges, sker følgende:

1. Stakken løftes på samme måde som beskrevet for [RCL] [Σ+].
2. Hældningskoefficienten *(A)* og skæringspunktet på Y-aksen *(B)* beregnes af:

$$A = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2} \qquad B = \frac{\Sigma y \Sigma x^2 - \Sigma x \Sigma xy}{n\Sigma x^2 - (\Sigma x)^2}$$

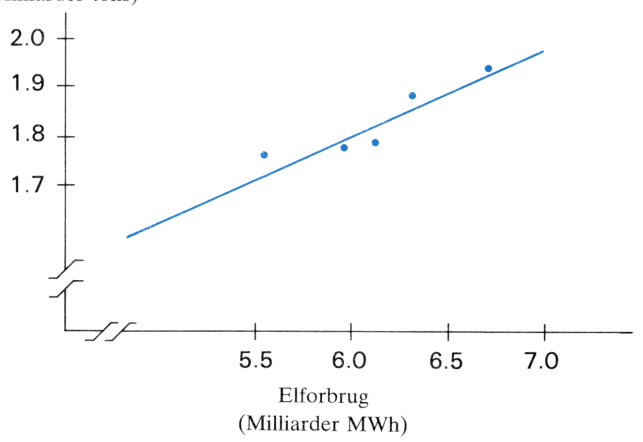
Hældningskoefficienten *(A)* lagres i Y-registret, og skæringspunktet *med Y-aksen (B)* lagres *i X-registret.*

[f] [L.R.]

T
Z
Y Hældningskoefficienten *(A)*
X Skæringspunktet på Y-aksen *(B)*

Eksempel. Beregn skæringspunktet på Y-aksen og hældningskoefficienten for den rette linie, der med mindste kvadraters metode findes at passe bedst til Voltz's data.

Løsning. Voltz *kunne* tegne punkterne ind i et rektangulært koordinatsystem og trække en ret linie efter at have vippet lidt frem og tilbage med en lineal. Men det er hurtigere at bruge [f] [L.R.] til beregning af A og B, som trods alt er de tal, hun er interesseret i at få frem. For slet ikke at tale om nøjagtighedsgevinsten.



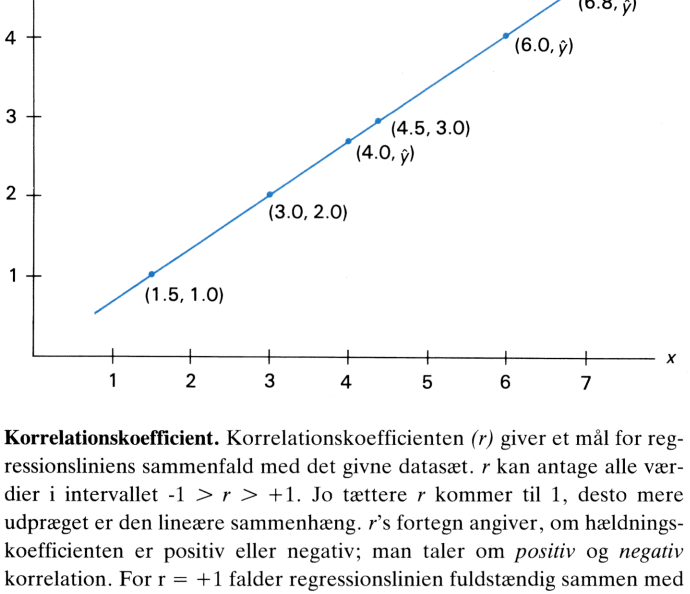
Indtastning	Lyspanel	
f L.R.	***0,7773***	Skæringspunkt på Y-akse.
$x \geqslant y$	***0,1718***	Hældningskoefficient.

Tallene i R_0-R_5 bruges i det næste eksempel.

Lineære estimater og korrelationskoefficient

Når f $\hat{y},r$ bruges, lagres et *lineært estimat* i X-registret, og *korrelationskoefficienten, r,* lagres i Y-registret.

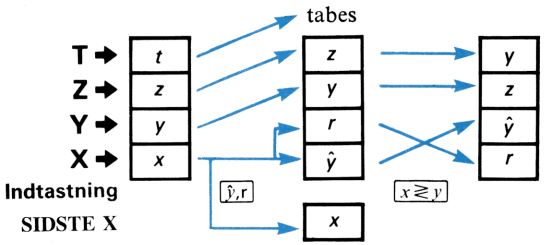
Lineære estimater. f $\hat{y},r$ beregner et *y*-estimat $(\hat{y})$ for tallet i X-registret *(x)*, når f $\hat{y},r$ bruges. Punktet *(x, ŷ)* er et punkt på *regressionslinien* $y = Ax + B$.



Korrelationskoefficient. Korrelationskoefficienten *(r)* giver et mål for regressionsliniens sammenfald med det givne datasæt. *r* kan antage alle værdier i intervallet $-1 > r > +1$. Jo tættere *r* kommer til 1, desto mere udpræget er den lineære sammenhæng. *r*'s fortegn angiver, om hældningskoefficienten er positiv eller negativ; man taler om *positiv* og *negativ* korrelation. For $r = +1$ falder regressionslinien fuldstændig sammen med

punkterne - dvs. at der ikke er forskel på y-estimater og y-værdier. Hvis $r = 0$, er der overhovedet ikke nogen sammenhæng mellem x- og y-værdierne; der er altså ikke antydning af en lineær sammenhæng mellem x- og y-værdierne.

I afsnit 2 blev det nævnt, at [g] [$\bar{x}$], [g] [s] og [f] [L.R.] ikke overfører X-registrets indhold til SIDSTE X. [f] [$\hat{y}$,r] bruger x, så her er der mening i at overføre X-registrets indhold til SIDSTE X. Dette sker da også, og stakken løftes én gang.



Eksempel. Voltz vil have korrelationskoefficienten bestemt for at afgøre, om der er en rimelig lineær sammenhæng mellem kulproduktion og elforbrug. Samtidig vil hun bestemme kulproduktionen i 1977 som en funktion af elforbruget i 1977.
På det tidspunkt, hvor Voltz arbejder med sin teori, er kulproduktionen for 1977 endnu ikke definitivt kendt (men formodes at ligge omkring 2 mia tons) – derimod kendes elforbruget ret nøje.

Indtastning	**Lyspanel**	
7.1417	***7,1417***	Elforbruget i 1977.
[f] [$\hat{y}$,r]	***2,0046***	Estimat for kulproduktion i 1977.
[$x \gtrless y$]	***0,9211***	Disse tal taler ikke imod Voltz's formodning om en lineær sammenhæng mellem kulproduktion og elforbrug.

Afsnit 4

Udlæsningsformater

Den kontinuerlige hukommelse sikrer, at HP-11C tænder med samme udlæsningsformat, som blev brugt, da der blev slukket for maskinen. Uanset hvordan tal repræsenteres i lyspanelet, vil lyspanelregistret altid indeholde en 10-cifret mantisse og en 2-cifret eksponent. Så når f.eks. udlæses som 3,1416, vil lyspanelregistret alligevel indeholde $3{,}141592654\times10^{00}$.

De ser kun de 5 første cifre, men også disse cifre ligger i HP-11C (afrundet til 4 decimaler).

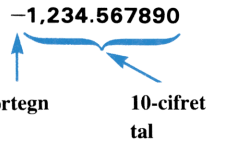
Udlæsningsformater

HP-11C kan udlæse tal i tre formater, der vælges med: f FIX, f SCI eller f ENG efterfulgt af et enkelt ciffer *(d)*, der specificerer formatet nærmere. Nedenfor vises, hvordan 123456 udlæses i de tre formater, når formateringscifret *(d)* er 4.

f FIX 4 :	123,456.0000	
f SCI 4 :	1.2346	05
f ENG 4 :	123.46	03

Fastkomma

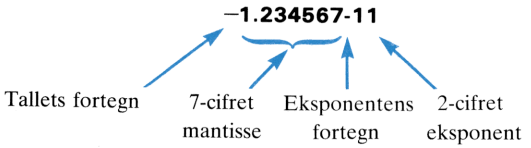
Med [f] [FIX] d ($0 < d < 9$) udlæses tal afrundet til et fast antal decimaler (d) og uden eksponent. Formatet har prioritering af de mest betydende cifre: hvis der er mere end 10-d cifre i tallets heltalsdel, skæres så mange decimaler væk, at heltalsdelen alligevel kan repræsenteres. Er der mere end 10 cifre i heltalstalen, skiftes automatisk til repræsentation med flydende tal [SCI] med samme d. Så snart tallet i X-registret igen har 10 eller færre cifre, skiftes tilbage til det oprindelige format.



Indtastning		Lyspanel	
123,45678 [ENTER]		***123,4568***	Tallet afrundes til 4 decimaler. Internt regnes dog med 10 cifre (hvor de to sidste er 00).
[f] [FIX]	6	***123,456780***	Tallet rundes op, hvis det første ikke-udlæste ciffer er 5 eller større.
[f] [FIX]	0	***123,***	
[f] [FIX]	4	***123,4568***	

Flydende tals repræsentation

Med [f] [SCI] d ($0 < d < 6$) udlæses tal som normaliserede flydende tal med ét ciffer foran decimalkommaet, afrunding til d cifre efter kommaet og en 2-cifret eksponent. d kan manuelt vælges som 7, 8 og 9, men der kan ikke udlæses mere end 6 cifre til højre for decimalkommaet.*



Indtastning	Lyspanel		
123.4567895 [ENTER]	***123,4568***		Fastkomma med 4 decimaler.
[f] [SCI] 2	***1,23***	***02***	Afrunding til $1{,}23\times10^2$ (første ikke-udlæste ciffer er mindre end 5).
[f] [SCI] 4	***1,2346***	***02***	Afrunding til $1{,}2346\times10^2$ (første ikke-udlæste ciffer er 5).
[f] [SCI] 6	***1,234568***	***02***	Det største antal cifre, der kan vises med flydende tals repræsentation.

Som eksemplet viser, afrundes resultaterne til d decimaler. Der kan ikke udlæses mere end 6 decimaler, men hvis man sætter d til 7, 8 eller 9, skubbes *afrundingen* til 7., 8. eller 9. ciffer.[§]

Indtastning	Lyspanel		
[f] [SCI] 7	***1,234567***	***02***	Tallet afrundes til 7 decimaler; med flydende tal vises højst 6 decimaler, så afrundingen ses ikke.
[f] [SCI] 8	***1,234567***	***02***	Afrunding til 8 decimaler.
[f] [SCI] 9	***1,234567***	***02***	Afrunding til 9 decimaler. Heller ikke med d=8 og d=9 ses nogen ændring.

Metrisk notation

Med [f] [ENG] d ($0 < d < 6$) udlæses tal i metrisk notation, en form for flydende tals repræsentation. d kan manuelt vælges som 7, 8 og 9, men der kan ikke udlæses mere end 6 cifre til højre for decimalkommaet.
Metrisk notation er kendetegnet ved at:

- Eksponenten er altid et multiplum af 3.
- d er antallet af cifre efter det første ciffer.

---

* [f] [SCI] 8 eller 9 samt [f] [ENG] 8 eller 9 bliver i programmer lagret som henholdsvis [f] [SCI] 7 og [f] [ENG] 7.

§ Hvis den del af tallet, der ikke udlæses, starter med et 9-tal, kan afrundingen forplante sig ind i lyspanelet, når man bruger [f] [SCI] 7 eller 8. F.eks. vil 1,00000094 ikke give afrunding med [f] [SCI] 7, hvorimod tallene 1,00000095 – 1,00000099 vil blive udlæst afrundet til 1,000001 med [f] [SCI] 7.

I metrisk notation bliver det mest betydende ciffer altid udlæst således, at d bliver antallet af cifre – ud over det første ciffer.

Indtastning		Lyspanel		
.012345		***0,012345***		
f ENG	1	***12,***	***-03***	Udlæsning i metrisk notation. Tallet afrundes til $d+1$ cifre, og eksponenten er et multiplum af 3.
f ENG	3	***12,35***	***-03***	Tallet udlæses afrundet til 4 cifre.
f ENG	6	***12,34500***	***-03***	
f ENG	0	***10,***	***-03***	Tallet afrundes til ét ciffer.

I det følgende eksempel ses, hvordan antallet af cifre foran kommaet automatisk justeres, så eksponenten forbliver et multiplum af 3.

Indtastning		Lyspanel		
f ENG	2	***12,3***	***-03***	Tallet fra sidste eksempel med f ENG 2.
10 ×		***123,***	***-03***	Kommaet skifter position.

Indtastning af eksponent

EEX bruges til at afslutte indtastning af et tals mantisse og gøre klar til indtastning af eksponenten. F.eks. indtastes Avogadros tal ($6{,}0222\times10^{26}$ kmol) således:

Indtastning	Lyspanel		
f FIX 4	***0,0123***		Skift til standardformat.
6.0222			
EEX	***6,0222***	***00***	00 markerer, at de følgende cifre betragtes som eksponentcifre.
26	***6,0222***	***26***	$6{,}0222\times10^{26}$.

Hvis eksponenten er éncifret, er det ikke nødvendigt at indtaste 2 eksponentcifre – enhver tast, der ikke bruges til talindtastning, vil afslutte eksponenten.

Et tal mindre end 1 har negativ eksponent. Sådanne tal indtastes ved at man indtaster mantissen, trykker på EEX og på fortegnsvenderen CHS og indtaster eksponenten. Når CHS bruges efter EEX virker fortegnsvendingen alene på eksponenten. F.eks. kan Plancks konstant ($6{,}6262\times10^{-34}$ Joule-sek) indtastes således:

Indtastning	Lyspanel		
6.6262 EEX	***6,6262***	***00***	
CHS	***6,6262***	***-00***	
3	***6,6262***	***-03***	
4	***6,6262***	***-34***	
ENTER	***6,6262***	***-34***	
50 ×	***3,3131***	***-32***	Joule-sek

Bemærk: Hvis de pladser, der er afsat til eksponenten, allerede er besat, vil eksponentcifrene overskrive disse – men cifrene bevares naturligvis internt.

Men EEX kan ikke aktiveres, hvis der er indtastet mere end 7 cifre foran kommaet, eller hvis tallet er et decimaltal med mere end 5 nuller foran det første betydende ciffer. I sådanne tilfælde må man selv flytte kommaet, så tallet kan indtastes. f.eks. kan $123456789{,}8\times10^{23}$ indtastes som $1234567{,}898\times10^{25}$, og $0{,}00000025\text{x}10^{-15}$ indtastes som $2{,}5\times10^{-22}$.

Mantisse

Alle tal, der lagres i HP-11C's rullestak eller dataregistre, repræsenteres med en 10-cifret mantisse og en 2-cifret eksponent. Hvis De på et vilkår-

ligt tidspunkt under et beregningsforløb ønsker at få udlæst alle mantissens cifre, gøres dette ved at trykke [f] **CLEAR** [PREFIX] og holde på [PREFIX], så længe mantissen ønskes udlæst. Når [PREFIX] slippes, skiftes tilbage til det oprindelige udlæsningsformat.

Indtastning	**Lyspanel**	
[f] [FIX] 4		Skift til standardformat.
[f] [π]	***3,1416***	π afrundet til 4 decimaler.
[f] CLEAR [PREFIX]	***3,141592654***	Alle 10 cifre udlæses, så længe der trykkes på

Afrunding af det 10. ciffer

Som tidligere nævnt lagres alle tal i HP-11C med 10 betydende cifre uanset, hvordan de udlæses. Også alle resultater afrundes til 10 cifre, inden de lagres i rullestak eller dataregistre. Dette gælder specielt for uendelige decimalbrøker som 2/3 = 0,66666666666... og transcendentale tal som = 3,1415926535..., tal, som HP-11C må repræsentere på endelig form. Det betyder bl.a., at der kan optræde små afrundingsfejl på 10. ciffer, og at disse fejl kan hobe sig op under omfattende beregningsforløb. I langt de fleste tilfælde vil fejlen dog ikke have nogen praktisk betydning. En nøjagtig vurdering af fejlkilder og af afrundingsfejlenes størrelse hører ind under den numeriske analyse og ligger uden for denne brugervejlednings rammer.

Kapitel 2

HP-11C
Programmering

Afsnit 1

Grundlæggende programmering

Hvad er et program?

Et program er en række manuelle indtastninger, som lagres i regnemaskinen. De lagrede indtastninger kan udføres eller køres lige så tit man vil – typisk ved at trykke på én eller to taster. Rullestakken opfører sig ens under programkørsel og manuel beregning og svaret, der udlæses efter programmets udførelse, er det samme som hvis beregningen var udført manuelt. Det er ikke nødvendigt at have kendskab til programmering for at lære at programmere HP-11C.

Hvorfor skrive programmer?

Programmer skrives for at spare tid. Så snart De har skrevet og lagret de indtastninger, der skal bruges til at løse et bestemt problem, behøver De ikke længere bruge tid på at tænke på de enkelte indtastninger og deres rækkefølge. Lommeregneren løser problemet. Og da det er meget simpelt at kontrollere et program i HP-11C, kan De have mere tiltro til resultaterne, end hvis de var fremkommet manuelt; De skal f.eks. ikke kontrollere om denne eller hin indtastning nu også var korrekt.
I det følgende gennemgås HP-11C's programmeringsfunktioner. I kapitel 3, afsnit 2 omtales nogle metoder, der gør det lettere at planlægge og udvikle overskuelige programmer.

Programkontrol

Automatisk splitning af hukommelsen

HP-11C holder selv styr på splitningen: fordelingen af den plads, der afsættes til henholdsvis data og program. Splitningen ændres automatisk

og påvirker hverken tallet i lyspanelregistret eller nogen tastfunktion. Derfor behøver De kun interessere Dem for følgende:

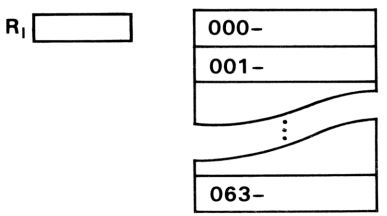
1. Forhold, der ændrer splitningen.
2. Hvad der sker, når splitningen ændres.

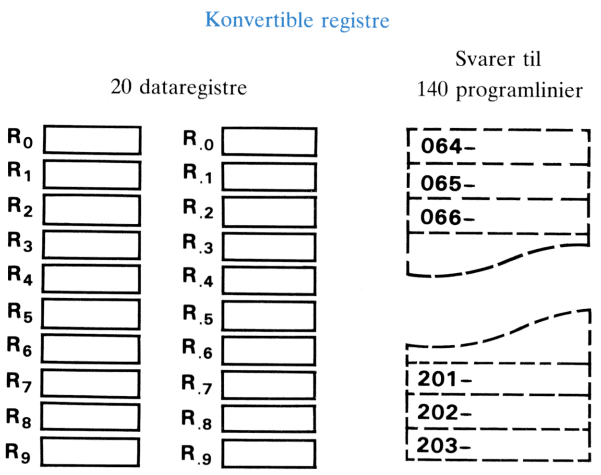
Når programhukommelsen eller hele hukommelsen nulstilles, er splitningen sat til 20 dataregistre (plus I-registret) og 63 programlinier.

Efterhånden som instruktioner indtastes, lagres de i den del af hukommelsen, der er afsat til program. Lagringen finder sted i programlinier, der hver indeholder én komplet instruktion. Hvis de 63 linier i den permanente programhukommelse fyldes op, vil HP-11C automatisk konvertere dataregister R_9 til 7 programlinier, når den næste instruktion indtastes. Herefter vil splitningen altså være: 20 dataregistre og 70 programlinier. Bruges også disse 7 nye programlinier, konverteres R_8 til 7 programlinier osv. indtil alle 20 konvertible dataregistre er lavet om til programlinier.

HP-11C's hukommelse

Permanente registre





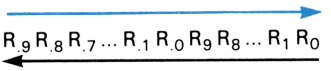
Hvis alle konvertible dataregistre er konverteret til programlinier, vil HP-11C have 203 programlinier og ét dataregister (I-registret). Nedenstående tabel viser sammenhængen mellem linienumre og dataregistre.

Korresponderende dataregistre og linienumre

$R_{.9}$	064—070	R_9	134—140
$R_{.8}$	071—077	R_8	141—147
$R_{.7}$	078—084	R_7	148—154
$R_{.6}$	085—091	R_6	155—161
$R_{.5}$	092—098	R_5	162—168
$R_{.4}$	099—105	R_4	169—175
$R_{.3}$	106—112	R_3	176—182
$R_{.2}$	113—119	R_2	183—189
$R_{.1}$	120—126	R_1	190—196
$R_{.0}$	127—133	R_0	197—203

Når programlinier slettes, tilbagekonverteres de løbende i blokke på 7 linier.

Rækkefølge for konvertering af dataregistre til programlinier

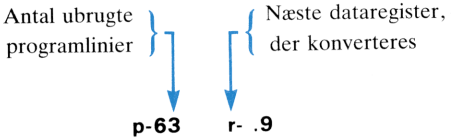


Rækkefølge for konvertering fra programlinier til dataregistre

I Tillæg C beskrives den automatiske splitning i nærmere detaljer.

[MEM]

Den aktuelle splitning kan udlæses med [g] [MEM]. Så længe der trykkes på [MEM], udlæses en kode, der angiver antallet af ubrugte programlinier, og hvilket dataregister der næste gang vil blive konverteret.



Resultat af [MEM], når programhukommelsen er tom

Tastkoder og linienumre

Når der skiftes til programmeringsstatus med [g] [P/R] og man indtaster instruktioner, bliver der udlæst koder for de enkelte indtastninger sammen med nummeret på den linie, der bruges til at lagre den komplette instruktion. Tastkoden vil være opbygget af én, to eller tre talgrupper afhængigt af, om den komplette instruktion er fremkommet ved brug af én, to eller tre taster. Talgrupperne består normalt af to cifre, der repræsenterer tastens placering på tastaturet; første ciffer er rækkenummeret, og andet ciffer er søjlenummeret, idet man betragter tastaturet som en tastmatrix. Af praktiske grunde har taltasterne éncifrede koder.

Linie nummer

017	42,	21,	11
Rækkenummer	4	2	1
Søjlenummer	2	1	1

Forkortet indtastning

I nogle tilfælde kan [f] udelades. F.eks. behøver man ikke trykke [STO] [f] [RAN#] ved lagring af et udgangstal, men kan nøjes med [STO] [RAN#]. Hvis man under programmering indtaster et sådant overflødigt [f], bliver det ikke registreret med tastkode.
Andre tilfælde af forkortet indtastning omtales, efterhånden som de bliver aktuelle.

Funktioner til programkontrol

Statusomskifter. Når man i normalstatus bruger [g] [P/R], skifter HP-11C fra normalstatus til programstatus og vice versa. Når maskinen er i programmeringsstatus, vises PRGM i lyspanelet, og programinstruktioner kan indtastes og slettes. I normalstatus kan man enten køre lagrede programmer eller udføre manuelle operationer.

Nulstilling af programhukommelsen. I *programmeringsstatus* slettes al information i programhukommelsen, og splitningen normaliseres til 21 dataregistre og 63 programlinier, når der trykkes på [f] CLEAR [PRGM]. I normalstatus sker der ikke andet, end at *programpilen* føres til linie 000. (Programpilen peger på den programlinie, der udlæses i programmeringsstatus – dvs. den som ville være under udførelse, hvis programmet kørte).

Hop til linie 000. Trykkes [GTO] [·] 000 føres programpilen til linie 000 i såvel normal- som programmeringsstatus.

Etiketter. HP-11C bruger alfaetiketterne [f] [LBL] [A] - [f] [LBL] [E] og de numeriske etiketter [f] [LBL] 0 - [f] [LBL] 9 til at markere forskellige nøglepunkter i programmer.
Ethvert program og underprogram indledes med en etiket. Desuden kan etiketter bruges til at markere linier, der skal hoppes hen til, når visse betingelser er opfyldt. Etiketterne fungerer således som *adresser* ved hop og kald af underprogrammer. I normalstatus kan et program, der begynder med en alfaetiket, aktiveres ved at trykke på [f] og den tast, der svarer til etiketten. De numeriske etiketter kan lige så vel indlede programmer, men bruges normalt til underprogrammer og som hopadresser. Programafsnit, der indledes med en numerisk etiket, kan aktiveres i

normalstatus hvis man trykker på [GSB] og den taltast, der svarer til etiketten.

Retur. [g] [RTN] vil som afslutning på et program stoppe programudførelsen og føre programpilen til linie 000. Som afslutning på et underprogram føres programpilen til linien umiddelbart efter den instruktion, der blev brugt til at kalde underprogrammet (se kapitel 2, afsnit 4).

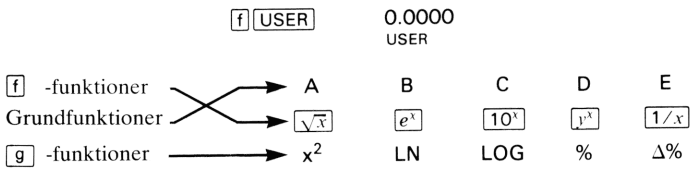
Kør/Stop. Når [R/S] mødes under programudførelse, stopper programmet, og kontrollen overgives til tastaturet, så man igen kan foretage manuelle beregninger. Trykkes der på [R/S], medens kontrollen er overgivet til tastaturet, starter programmet fra den programlinie, som programpilen peger på.

(Sprogbrugen er noget løs; *»programmet«* betegner både instruktionerne og den mekanisme, der udfører instruktionerne – man taler således om et program, der starter, kører, stopper og møder instruktioner).

Pause. Når [f] [PSE] mødes som instruktion i et program, standses programudførelsen i ca. ét sekund, medens X-registrets indhold udlæses. Herefter genoptages programudførelsen automatisk. Der kan lægges flere pauseinstruktioner efter hinanden, hvis man får brug for det.

USER-status

USER-status er en speciel status, som sparer indtastninger under brug af programmer. I USER-status er der byttet om på placeringen af grundfunktionerne og de alternative [f]-funktioner i tastaturets øverste række.



Det bliver herved muligt, i normalstatus, at udføre ethvert program, der starter med en af etiketterne A-E, ved blot at trykke på en enkelt tast – dvs. uden først at skulle trykke på [f]. Det kan synes som en overflødig

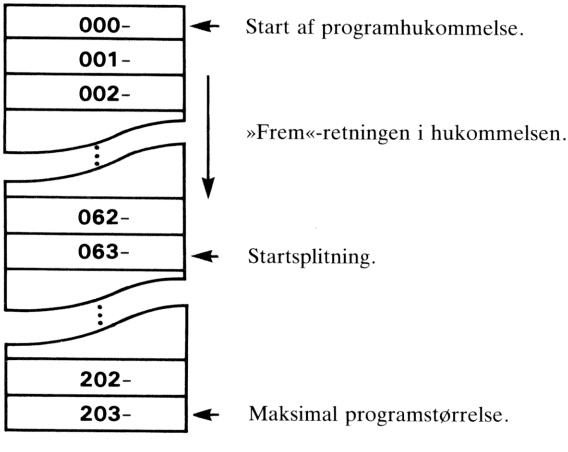
finesse – men specielt under indlæsning af mange data eller valg mellem flere programmer, der er lagret samtidigt, er det en stor fordel at kunne bruge de fem øverste taster som *brugerdefinérbare taster.*

Bemærk: For at undgå fejlprogrammering eller utilsigtet start af et program, bør man kun skifte til USER-status, når der er brug for det.

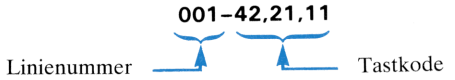
Programhukommelsen

Som De måske husker fra varmetabsprogrammet i starten af håndbogen, bruger man de samme indtastninger til programmeret løsning af en opgave, som når man løser opgaven manuelt. Instruktionerne lagres i *programhukommelsen.* Tryk [GTO] [·] 000, der placerer programpilen ved programhukommelsens start: linie 000. Hvis De ikke allerede har gjort det, skal De nu trykke [g] [P/R], så der skiftes til programmeringsstatus. (Husk, at når HP-11C er i programmeringsstatus, vises PRGM i lyspanelet). I lyspanelet står der nu 000 – som tegn på, at programpilen står ved starten af programhukommelsen.
Programhukommelsen er uafhængig af rullestakken, SIDSTE X, R_I, og af de dataregistre, der ikke er konverteret til programlinier.

Programhukommelse



Når HP-11C er i programmeringsstatus, vil tallet i venstre side af lyspanelet angive *linienummeret*. Tryk [f] CLEAR [PRGM] og derefter [f] [LBL] [A] – den første instruktion i varmetabsprogrammet. Herefter vil lyspanelet se således ud:



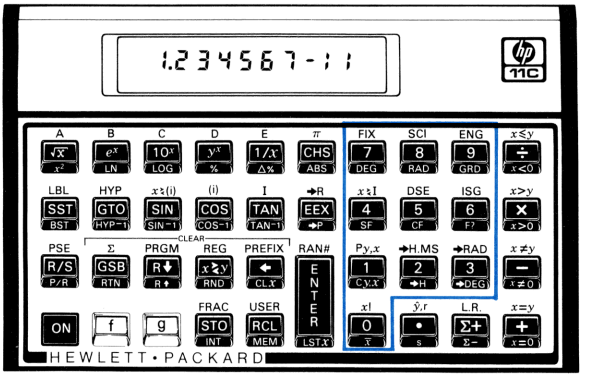
Tallet 001 viser, at programpilen peger på linie 001. De andre tal i lyspanelet er tastkoder for de instruktioner, der er lagret i linie 001. Indtast dernæst 8. Nu ser lyspanelet således ud:

Koden viser, at tallet 8 er lagret som instruktion nr. 2.

Hver linie i programhukommelsen kan indeholde en programinstruktion – uanset om der skal bruges én, to eller tre indtastninger til at aktivere instruktionen. F.eks. vil [CHS] (der skifter fortegn på tallet i lyspanelregistret) kunne lagres med en enkelt indtastning, hvorimod [STO] [+] 6 (der lægger tallet i lyspanelregistret til tallet i R_7) kræver 3 indtastninger. Men i begge tilfælde bruges én programlinie.

Fortolkning af tastkoder

De fleste af HP-11C's tastkoder svarer nøje til tasternes placering i tastaturets rækker og søjler. Rækker er nummererede 1 til 4, medens søjlerne er nummererede 1 til 10, hvor den tiende kolonne repræsenteres ved 0. Det betyder således, at sinus-tasten, der er anbragt i 2. rækkes 3. søjle, har tastkoden 23, og at plus-tasten, der er anbragt i 4. rækkes 10. søjle, har tastkoden 40. Taltasterne er de eneste taster, der ikke følger dette system; de repræsenteres ganske enkelt ved det tal, der er trykt på den enkelte tast.



Encifrede tastkoder

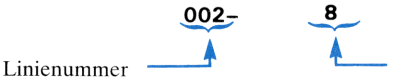
En programlinie kan bestå af én til tre tastkoder:

	053– 23	054– 43 33		055–44,40, 5		
	SIN	g	R↑	STO	+	5
række	2	4	3	4	4	»5«
søjle	3	3	3	4	10	taltast

Lad os igen se på den programinstruktion, der netop blev indtastet. Tryk g BST , så den foregående linie bliver udlæst:

001- viser, at der er tale om programlinie nr. 1. Det næste cifferpar, 42, repræsenterer f (4. rækkes 2. søjle); 21 repræsenterer LBL (2. rækkes 1. søjle; 11 repræsenterer A . På denne måde bliver alle programmérbare funktioner (undtagen de, der aktiveres af en taltast) repræsenteret ved hjælp af en tocifret tastkode.

Lad os se på endnu et eksempel. Tryk på SST , så den næste instruktion i varmetabsprogrammet bliver udlæst.



I dette tilfælde er »8« det eneste tal udover linienummeret, så tastkoden står for *tallet* 8. Til højre vises de forskellige kombinationer, hvor taltasten »8« indgår.

Taster		Tastkode
f SCI	→	42 8
8	→	8
g RAD	→	43 8

Resten af varmetabsprogrammet vises nedenfor. Indtast programmet og kontrollér tastkoderne efter hver indtastning.

Indtastning	Lyspanel		
×	***003-***	***20***	2. rækkes 10. søjle.
2	***004-***	***2***	»2«-tasten.
·	***005-***	***48***	4. rækkes 8. søjle.
6	***006-***	***6***	»6«-tasten.
7	***007-***	***7***	»7«-tasten.
×	***008-***	***20***	2. rækkes 10. søjle.
g RTN	***009-***	***43 32***	Afslutter programmet.
g P/R			Ophæver programmeringsstatus.

Programmering

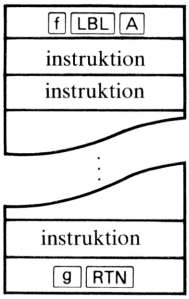
Det foregående emne, tastkoder, handlede om de enkelte instruktioner. Vi vil nu bruge lidt tid på en detaljeret gennemgang af en egentlig programmeringsproces. Til det formål vil vi lave et lille program, der kun tjener til at illustrere de enkelte trin i programmeringsprocessen.
En cirkels areal kan bestemmes af formlen $A = \pi r^2$, hvor man først indtaster radius, r, kvadrerer denne med g x^2, henter π med f π og trykker på × for at gange tallen med hinanden.
Opgaven løses altså med følgende tastsekvens:

g x^2
f π
×

Indledning og afslutning af program

Indledningen (starten) af et program markeres med en *etiket* af formen f LBL *etiket,* hvor etiketten er et af bogstaverne A-E eller tallene 0-9. Brugen af etiketter gør det muligt dels at have mere end ét program ad gangen i programhukommelsen, dels at udføre disse programmer i vilkårlig rækkefølge.

Programhukommelse



LBL indleder programmet.

RTN afslutter programmet.

Afslutningen på et program markeres med g RTN. Når RTN bruges på denne måde, vil programpilen hoppe til linie 000, når den møder RTN, hvorefter kontrollen overgives til tastaturet.

Bemærk: Hvis programpilen sættes til at gå forbi den sidste linie inden for den givne splitning, fortolker den automatisk denne ordre som g RTN og hopper op til linie 000.

Det fuldstændige program

Programmet til beregning af cirklers areal vil derefter se således ud:

f LBL A	Markerer starten på programmet og giver det navn.
g x^2	Kvadrerer radius, som er indtastet.
f π	Lagrer π i X-registret (løfter stakken).
×	Beregner produktet af π og kvadratet på radius (stakken falder).
g RTN	Markerer afslutningen på programmet; programpilen returnerer til linie 000, og kontrollen overgives til tastaturet.

Lagring af programmet

Programmet kan lagres før eller efter et program, der allerede er lagret i programhukommelsen. Hvis det nye program skal lagres før det eksisterende, starter man fra linie 000 og indtaster instruktionerne. Hver gang man indtaster en ny instruktion, bliver hver eneste efterfølgende instruktion i programmet skubbet én linie frem.

Der gøres klar til indtastning af cirkelarealprogrammet på følgende måde:

1. Tryk g P/R, så der skiftes til programmeringsstatus. I lyspanelet vises **PRGM** -indikatoren.

2. Tryk [f] CLEAR [PRGM], så alt, der er lagret i programhukommelsen, bliver slettet. (hvis De vil bevare den information, der allerede er lagret i hukommelsen, skal De i stedet trykke [GTO] [·] 000, der blot sætter programpilen op til linie 000).

I lyspanelets venstre side vises 000- som tegn på, at programpilen står ud for starten af programhukommelsen.
Programmet består af instruktionerne:
Opgaven løses altså med følgende tastsekvens:

[f] [LBL] [A]
[g] [x^2]
[f] [π]
[×]
[g] [RTN]

Tryk nu på den første tast i programmet: [f].

Indtastning	Lyspanel
	000-

De kan se, at indholdet af lyspanelet ikke har ændret sig. Det sker først, når hele instruktionen er indtastet:

Indtastning	Lyspanel	
[f] [LBL]	***000-***	
[A]	***001-42, 21, 11***	[f] [LBL] [A] er nu lagret i programhukommelsen.

Når linienummeret i lyspanelet skifter, er det tegn på, at den instruktion, hvis tastkode nu udlæses, er blevet lagret i programhukommelsen. Husk, at intet lagres, før hele instruktionen er indtastet (uanset om der kræves én, to eller tre indtastninger).
Indtast nu resten af programmet.

Indtastning	Lyspanel	
[g] [x^2]	***002-***	***43 11***
[f] [π]	***003-***	***42 16***
[×]	***004-***	***20***
[g] [RTN]	***005-***	***43 32***

Programmet, der kan bestemme cirkelarealer, er hermed lagret i programhukommelsen.

Kørsel af programmet

Programmer kan kun udføres i normalstatus, så inden De afprøver pro-

grammet, skal De først trykke [g] [P/R], der skifter fra programmerings- til normalstatus.

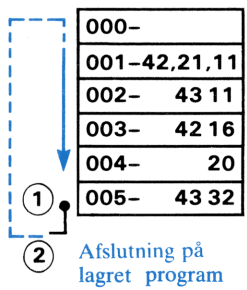
Når et program skal køres, må man først indtaste de nødvendige data og trykke på [f] efterfulgt af programmets etiket (A-E).
I dette tilfælde skal De altså indtaste en radius og trykke [f] [A] .

Eksempel. Beregn arealerne for cirklerne med radier på 7,5 cm, 9 tommer og 15,3 meter.

Indtastning	**Lyspanel**	
7.5 [f] [A]	***176,7146***	Kvadratcentimeter.
9 [f] [A]	***254,4690***	Kvadrattommer.
15.3 [f] [A]	***735,4154***	Kvadratmeter.

Søgning efter etiketter. Da HP-11C blev sat i normalstatus, stod programpilen ud for linie 005 (den sidste linie, der blev brugt til lagring af en instruktion). Efter [f] [A] begynder maskinen en *søgning* efter etiket A. Det betyder, at programpilen starter fra sin øjeblikkelige position (linie 005) og bevæger sig linievist frem i programmet, indtil den finder en linie med instruktionen [f] [LBL] [A]. Først når det sker, starter udførelsen af det lagrede program med etiketten A.

Linie 005 indeholder ikke [f] [LBL] [A], og linie 006 indeholder ikke nogen programinstruktion, så pilen hopper op til linie 000 og fortsætter søgningen derfra. I linie 001 finder den [f] [LBL] [A] og starter programkørslen.



Udførelse af programinstruktioner. Lommeregneren udfører instruktionerne i den rækkefølge, de er indtastet. Dvs. at [g] [x^2] i linie 002 udføres først, dernæst [f] [π] i linie 003 osv. indtil [g] [RTN] eller [R/S] udføres, eller indtil programpilen møder tomme linier eller når afslutningen af den tilgængelige programhukommelse.
I dette tilfælde er der en [g] [RTN] -instruktion i linie 005, så her hopper programpilen op til linie 000 og stopper. Resultatet af beregningen, tallet

i X-registret, bliver derved udlæst. Under udførelsen af længere programmer vises **running** blinkende i lyspanelet.

Ikke-programmérbare funktioner. Næsten alle HP-11C's funktioner kan lagres som instruktioner i et program. Men de følgende funktioner er *styre*funktioner, som det ikke giver nogen mening at lagre som programinstruktioner – så de kan ikke lagres i programhukommelsen.

f CLEAR PRGM	g P/R	SST
f CLEAR PREFIX	g MEM	←
GTO · *nnn*	· / ON	g BST
f USER		

Brug af USER-status

Vi vil nu vælge USER-status for at gøre det endnu lettere at køre cirkelarealprogrammet. Samtidig anskueliggøres virkningerne af USER.

Indtastning	Lyspanel	
f USER	***USER***	Valg af USER-status. USER vises i lyspanelet.
7.5 A	***176,7146***	I USER-status bliver A-E de primære tastfunktioner for tasterne i øverste række
9 A	***254,4690***	
15.3 A	***735,4154***	
4 f $\sqrt{x}$	***2,0000***	
1 f e^x	***2,7183***	
1 f 10^x	***10,0000***	I USER-status bliver grundfunktionerne i første række til alternative f -funktioner.
2 ENTER	***2,0000***	
8 f y^x	***256,0000***	
· 5 f $1/x$	***2,0000***	
f USER	***2,0000***	Ophæver USER-status.

Stop og pauser under programkørsel

Det kan være hensigtsmæssigt at lægge stop ind i et program, så der kan indtastes nye data, efter at de først indtastede er behandlet. I andre tilfælde vil man blot have udlæst et mellemresultat og har ikke brug for at indtaste data, så det er tilstrækkeligt med et tidsbegrænset stop med automatisk genstart. HP-11C har to funktioner, der passer til disse situationer: R/S *(kør/stop)* og f PSE *(pause)*.

Programmerede stop

[R/S] *(kør/stop)* kan dels bruges som programinstruktion, dels som funktion aktiveret fra tastaturet.

Når [R/S] bruges fra tastaruret, sker følgende:

1. Hvis et program kører, stoppes kørslen.
2. Hvis et program er standset (f.eks. af en [g] [RTN] - eller [R/S] -instruktion eller af tryk på [R/S]), starter kørslen med linien efter den programlinie, som programpilen peger på.
   (Hvis man i normalstatus trykker på [R/S] og holder den nede, vil programpilen gå frem til næste linie, og lyspanelet udlæser dens nummer og tastkode. Når tasten slippes, starter udførelsen af programmet fra denne linie).

[R/S] kan sættes ind i et program, hvor man ønsker at kunne indtaste nye data. Når programpilen møder [R/S], overgives kontrollen til tastaturet. Efter indtastning af data startes kørslen med tryk på [R/S].

Eksempel: Universal Tin er et firma, der fremstiller konservesdåser. De har brug for hurtig og nøjagtig beregning af cylindriske dåsers volumen, og af hensyn til stablingsberegninger skal man også kende grundfladens areal.

Følgende program beregner først arealet af en cylinders grundflade. Derefter beregnes voluminet efter tryk på [R/S].

Formlen er:

$$\text{Volumen} = \text{grundflade} \times \text{højde} = \pi r^2 \times h$$

Inden programmet startes, skal radius (r) og højden (h) af dåsen indtastes i henholdsvis X- og Y-registrene.
Programmet lagres ved at skifte til programmeringsstatus med [g] [P/R] og derefter indtaste nedenstående instruktioner:

Indtastning	Lyspanel	
[f] CLEAR [PRGM]	***000-***	Sletning af program-hukommelsen.
[f] [LBL] [A]	***000- 42, 21, 11***	Programmet kaldes »A«.
[g] [x^2]	***002- 43 11***	Kvadrering af radius.
[f] [π]	***003- 42 16***	π lægges i X-registret.
[×]	***004- 20***	Grundfladens areal.
[R/S]	***005- 31***	Stop programmet, så grundfladens areal bliver udlæst.
[×]	***006- 20***	Beregning af voluminet.
[g] [RTN]	***007- 43 32***	Afslutning på program.

Tryk [g] [P/R], så HP-11C kommer i normalstatus. Udfyld dernæst følgende tabel:

Højde	Radius	Grundfladens areal	Volumen
25	10,0	?	?
8	4,5	?	?

Indtastning	Lyspanel	
25 [ENTER]	***25,0000***	Højden indtastes og lagres i Y-registret.
10 [f] [A]	***314,1593***	Radius indtastes, og lagres i X-registret, og programmet startes; det stopper efter beregning af grundfladens areal.
[R/S]	***7.853,9816***	Udlæsning af voluminet.

Indtastning	Lyspanel	
8 [ENTER]	***8,0000***	Højden indtastes og lagres i Y-registret.
4.5 [f] [A]	***63,6173***	Radius indtastes og lagres i X-registret, og programmet startes; det stopper efter beregning af grundfladens areal.
[R/S]	***508,9380***	Udlæsning af det andet volumen.

Når højden er lagret i Y-registret, og radius er lagret i X-registret, vil brug af [f] [A] i normalstatus beregne arealet af dåsens grundflade. Programmet stopper, når det møder [R/S] og fortsætter først, når [R/S] bruges. Herefter beregnes og udlæses voluminet, og programpilen returnerer til linie 000.
I kapitel 3, afsnit 2 diskuteres forskellige metoder til rationel indlæsning af data.

Programmerede pauser

Når programmet møder [f] [PSE] stopper kørslen, og tallet i X-registret udlæses i ca. 1 sekund, inden kørslen automatisk genoptages. Har man brug for en længere varende udlæsning, kan der sættes flere pauseinstruktioner efter hinanden.
For at se hvordan [f] [PSE] bruges i et program, vil vi modificere cylinderprogrammet således, at programmet ikke stopper efter beregning af grundfladens areal, men blot udlæser resultatet i ca. 1½ sekund, inden programmet fortsætter med beregning af voluminet.
Programmet lagres ved at man skifter til programmeringsstatus med [g] [P/R] og derefter indtaster nedenstående instruktioner:

Indtastning	Lyspanel	
[f] CLEAR [PRGM]	***000-***	Sletning af programhukommelsen.
[f] [LBL] [A]	***000- 42, 21, 11***	Programmet kaldes »A«.
[g] [x^2]	***002- 43 11***	Kvadrering af radius.
[f] [π]	***003- 42 16***	π lægges i X-registret.

Indtastning	Lyspanel		
[×]	*004-*	*20*	Grundfladens areal.
[f] [PSE]	*005-*	*42 31*	Pause, hvor grundfladens areal udlæses i ca. et sekund.
[×]	*006-*	*20*	Beregning af voluminet.
[g] [RTN]	*007-*	*43 32*	Afslutning på program.

Også dette program forudsætter, at højden er lagret i Y-registret, og at radius er lagret i X-registret, når programmet startes.
Når De har lagret instruktionerne, skal De trykke [g] [P/R] og udfylde følgende tabel.

Højde	Radius	Grundfladens areal	Volumen
20	15	?	?
10	5	?	?

Indtastning	Lyspanel	
20 [ENTER]	*20,0000*	Højden indtastes og lagres i Y-registret.
15 [f] [A]	*706,8583*	Radius indtastes og lagres i X-registret, og programmet startes; grundfladens areal udlæses i ca. 1 sekund.
	14.137,1669	Stop af program og udlæsning af voluminet.
10 [ENTER]	*10,0000*	Højden indtastes og lagres i Y-registret.
5 [f] [A]	*78,5398*	Radius indtastes og lagres i X-registret, og programmet startes; grundfladens areal udlæses i ca. 1 sekund.
	785,3982	Stop af program og udlæsning af det andet volumen.

Uventede programstop

Det vil undertiden ske, at programmet stopper på grund af en fejl i programmet. Følgende oversigt skulle gøre det muligt for Dem at afsløre en række af de mest almindelige årsager til sådanne uventede programstop.

Udførelse af [g] [RTN]. Hvis [g] [RTN] mødes, og instruktionen ikke afslutter et *under*program, vil programpilen hoppe til linie 000, hvorefter programudførelsen stopper.

Kørsel til sidste linie i programhukommelsen. Hvis den sidste instruktion i programhukommelsen ikke er [GTO], [GSB] eller [g] [RTN] som afslutning på et underprogram, vil programpilen hoppe til linie 000, hvorefter programmet stopper.

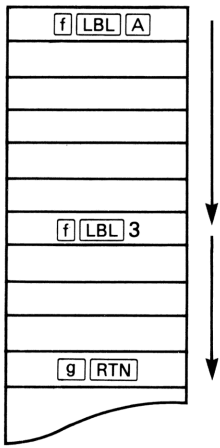
Tryk på en tast under programudførelse. Tryk på en hvilken som helst tast vil stoppe programudførelsen. Den interne logik sikrer, at De ikke kan stoppe programudførelsen midt under indtastning af rent numeriske instruktioner (f.eks. en konstant, der er lagret i programhukommelsen). Hvis De trykker på en tast, medens programmet indlæser et tal til X-registret, fortsættes talindlæsningen, og programmet udfører den næste ikke-numeriske instruktion, inden programudførelsen stopper.
Når et program er standset på denne måde, kan det genstartes med [R/S], hvis HP-11C er i normalstatus, og De iøvrigt ikke har rørt nogen tast efter stoppet. I de fleste tilfælde kan De udmærket foretage beregninger under et stop, men hvis tallene i rullestakken eller stakløftstatus ændres, kan der komme højst uventede resultater ud af den fortsatte kørsel.

Fejlstop. Hvis programmet medfører udførelsen af en fejlgivende operation (Se Tillæg A), vil programudførelsen straks stoppe, og i lyspanelet udlæses fejlmeddelelsen **Error** efterfulgt af et ciffer, der nærmere specificerer fejlens karakter. Fejlmeddelelsen ophæves ved at trykke på en vilkårlig tast. Ved at skifte til programmeringsstatus kan De derefter få opklaret, i hvilken linie fejlen blev *opdaget* – hvilket ikke nødvendigvis er linien med den instruktion, der *forårsagede* fejlen.
Registeraritmetik, der ville medføre overløb i et dataregister, resulterer i fejlmeddelelsen **Error 1** og stop af programmet. Tallet i det dataregister, der skulle bruges i forbindelse med registeraritmetikken, påvirkes ikke af fejltilstanden og har således sin oprindelige værdi. Når fejlmeddelelsen slettes, genudlæses X-registrets oprindelige værdi.

Hvis en beregning giver et resultat, der er mindre end 1×10^{-99}, indsættes 0 for tallet, hvorefter beregningen fortsætter.

Etiketter

Som tidligere nævnt bruges etiketter som adresser – de fortæller maskinen, hvor et program skal begynde eller fortsætte. Bemærk, at etiketterne ignoreres under programkørslen. Nedenfor vises en programstump, hvor [f] [A] vil starte programmet fra [f] [LBL] [A]. Derfra fortsætter det frem gennem programhukommelsen og møder bl.a. [f] [LBL] 3. Denne etiket ignoreres totalt; den aktuelle instruktion er ikke en besked om at lede efter etiketten »3«. Først når [g] [RTN] mødes, hopper programpilen op til 000 og stopper programmet.



Når der trykkes på [f] [A], starter programmet her.

Der er ikke nogen [R/S] eller [g] [RTN] inden [f] [LBL] 3, så programmet fortsætter uanfægtet...

... til [g] [RTN], hvorefter programpilen hopper til linie 000, og programmet stopper.

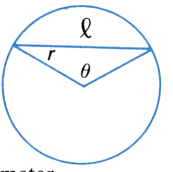
Opgaver

1. De skal skrive og lagre et program, der omregner temperaturer fra Celcius til Fahrenheit. De kan bruge formlen °F = 1,8°C + 32.
   Kald programmet »A« og afslut med [g] [RTN].
   Afprøv programmet på 40°C, 0°C og 72°C.

   Svar: -40,0000°F, 32,0000°F og 161,6000°F

2. Lav et program, der beregner længden af korden I over vinklen θ i en cirkel med radius r. De kan bruge formlen $I = 2r \sin \theta/2$.
Programmet laves, så data indtastes i rækkefølgen: r, θ.
Kald programmet »B« og udfyld nedenstående tabel.

r (meter)	θ	ℓ
25	30	?
50	45	?
100	90	?



Svar: 12,9410 meter, 38,2683 meter, 141,4214 meter.

3. Bestem tastkoderne for følgende operationer:
[g] [%], [f] [→RAD], [STO] [+] 1, [SIN], [f] [LBL] [D] og [STO] [RAN#] *.

Svar: 43 14; 42 3; 44, 40, 1; 23; 42, 21, 14; 44 36*.

4. Hvor mange programlinier skal der bruges til lagring af nedenstående programafsnit?

a. 2 [ENTER] 3 [+].
b. 10 [STO] 6 [RCL] 5 [×].
c. 100 [STO] 1 50 [STO] [×] 1 [RCL] 2 [f] [π] [÷].

Svar a. 4; b. 5; c. 10.

*) Se side 74, Forkortet indtastning.

Afsnit 2

Editering

Selv den mest erfarne programmør vil af og til begå fejl under programmeringen. Det kan dreje sig om fejl i den formel, der bruges, logiske fejl ved formuleringen af programmet eller simple indtastningsfejl. HP-11C har en række funktioner, der letter rettearbejdet – eller editeringen, som det normalt kaldes.

Programfejl og kontinuerlig hukommelse

HP-11C's kontinuerlige hukommelse bevarer alle data i rullestakken, SIDSTE X, dataregistrene og programhukommelsen samt information om f.eks. vinkelmål og flagstatus (flag omtales senere).
Af den grund kan det være hensigtsmæssigt at gøre visse programmer *selv-initialiserende*. Et selv-initialiserende program indeholder alle de programinstruktioner, der skal til for at vælge den rette statuskombination og i det hele taget sikre, at beregninger, der foretages mellem programkørsler, ikke får betydning for programmets resultater.

Fejlsøgning

Inden et program bruges til noget alvorligt, bør man altid kontrollere det med et sæt testdata. Disse data skal være valgt således, at man får afprøvet programmet dels med nogle tal, der giver klart gennemskuelige resultater, dels i grænserne for anvendelse og nøjagtighed. I de fleste tilfælde vil det desuden være rimeligt at teste programmet med data, som man véd skal give en fejlmelding. Selv-initialiserende programmer bør afprøves med meningsløse data i de dataregistre, der skal initialiseres og med vilkårligt valgt vinkelmål osv.
Editeringsfaciliteterne på HP-11C giver Dem hurtig og let adgang til alle dele af et program, uanset om De skal editere, kontrollere eller dokumentere. Hvis et program stopper under kørslen, kan De notere Dem fejlkoden, slette denne og trykke [g] [P/R], hvorefter den linie, hvor fejlen

blev opdaget, bliver udlæst. Hvis De har en mistanke om, at en del af programmet er fejlbehæftet, kan programafsnittet udføres linie for linie, så De hele tiden kan overvåge udførelsen. Derefter kan editeringsfunktionerne bruges til at rette eventuelle fejl.

Editeringsfunktioner

HP-11C har fire ikke-programmérbare editeringsfunktioner, der bruges til at føre programpilen rundt i et program og til at rette de enkelte programinstruktioner:

[SST] [g] [BST] [GTO] [.] *nnn* og [←]

SST

[SST] står for Single STep – enkelttrin – og er en funktion, der fører programpilen én linie frem.

I programmeringsstatus:
Når man trykker på [SST], går programpilen frem til næste linie og udlæser indholdet. Tasten er repeterende; dvs. så længe den holdes nedtrykket, vil programpilen gennemløbe programmet linie for linie.
[SST] udfører ikke programinstruktionerne, når HP-11C er i programmeringsstatus.

001-42, 21, 11
[SST]
002- 41 13

I normalstatus:
Når der trykkes på [SST], rykker programpilen frem til næste linie, hvis indhold udlæses. Når man slipper tasten, udføres linien, og resultatet af instruktionen udlæses.

BST

[BST] står for Back STep – og er en funktion, der fører programpilen én linie tilbage. Tasten har samme repeterende effekt som [SST]. Instruktionerne i de programlinier, der udlæses, bliver ikke udført.

002- 43 13
[g] [BST]
001-42, 21, 11

HOP til linie ***nnn***. Efter [GTO] [.] *nnn* hopper programpilen hen til linie *nnn*. Dette gælder både i normalstatus og i programmeringsstatus. I normalstatus vil [GTO] efterfulgt af en alfa/numerisk etiket føre programpilen hen til første forekomst af etiketten.

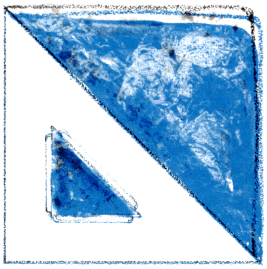
Slettetast. Når [←] bruges i programmeringsstatus, slettes den linie, der netop udlæses. Alle programinstruktioner, der står efter den slettede instruktion, vil automatisk rykke én linie opad og få nyt linienummer. I normalstatus påvirker [←] ikke programhukommelsen, men kun X-registret.

Brugen af [←]

	Før De trykker [←]:		Efter De har trykket [←]:	
(Lyspanel)	001-42,21,11	→	001-42,21,11	(Lyspanel)
	002- 43 13		002- 23	
	003- 23	→	003- 45 0	
	004- 45 0	→	004- 45 1	
	005- 45 1			

Eksempel på editering

Indtast det følgende program, som beregner hypotenusen i en retvinklet trekant med kendte kateter. Programmet skal bruges til at illustrere editeringsmekanismerne.



Siden c bestemmes ved hjælp af $c = \sqrt{a^2 + b^2}$, hvor a og b er kateterne. a lagres i Y-registret, og b lagres i X.

Brug [g] [P/R] til at skifte til programmeringsstatus.

Indtastning	Lyspanel		
[f] CLEAR [PRGM]	*000-*		Sletter program-hukommelsen.
[f] [LBL] [E]	*001-*	*42, 21, 15*	Programmet kaldes »E«.
[g] [x^2]	*002-*	*43 11*	Kvadrerer b.
[$x \geqslant y$]	*003-*	*34*	Lagrer b^2 i Y og henter a frem til X.
[g] [x^2]	*004*	*43 11*	Kvadrerer a.
[+]	*005*	*40*	$a^2 + b^2$.
[$\sqrt{x}$]	*006-*	*11*	$\sqrt{a^2 + b^2}$
[g] [RTN]	*007-*	*43 32*	Afslutter programmet.

Tryk [g] [P/R], så HP-11C skifter til normalstatus.
Programmet testes ved at beregne hypotenusen i en retvinklet trekant med $a = 22$ m og $b = 9$ m.

Indtastning	Lyspanel	
22 [ENTER]	*22,0000*	Indtastning af a.
9	*9,0000*	Indtastning af b.
[f] [E]	*23,7697*	c i meter.

SST-udførelse af program

I længere programmer vil man sjældent direkte kunne gennemskue en fejlmeddelelse; programmet stopper ikke ved den instruktion, der giver anledning til fejlen men ved den linie, hvor fejlen gør det umuligt for maskinen at fortsætte. I sådanne tilfælde kan man bruge [SST] til at udføre programmet linie for linie med maskinen i normalstatus.

Selvom det lagrede program synes at være rigtigt, vil vi nu udføre det med [SST]. [SST]-udførelse starter altid med den linie, som programpilen peger på. Da [g] [RTN] medfører, at programpilen hopper op til linie 000, kan vi i dette tilfælde nøjes med at indtaste startværdierne og straks begynde at udføre programmet med [SST].
Tryk på [SST], hold tasten nede et øjeblik, så linienummeret og tastkoden bliver udlæst og slip så tasten. Nu udføres operationen, resultatet udlæses, og programpilen går videre til næste linie. Processen gentages, indtil man har udført alle linier i det programafsnit, man ønsker at kontrollere.

Indtastning	Lyspanel	
22 [ENTER]	***22,0000***	Indtastning af *a*.
9	***9,0000***	Indtastning af *b*.
[SST]	***001- 42, 21, 15***	Når [SST] holdes nedtrykket, udlæses indholdet af linie 001.
	9,0000	Når tasten slippes, udføres instruktionen.
[SST]	***002- 43 11***	Tastkoden for [g] [x^2].
	81,0000	[SST] slippes, og [g] [x^2] udføres.
[SST]	***003- 34***	Tastkoden for [$x \geq y$].
	22,0000	[$x \geq y$] udføres.
[SST]	***004- 43 11***	
	484,0000	[g] [x^2] udføres.
[SST]	***005- 40***	
	565,0000	[+] udføres.
[SST]	***006- 11***	
	23,7697	[$\sqrt{x}$] udføres.
[SST]	***007 43 32***	
	23,7697	[g] [RTN] udføres. Programudførelsen afsluttes, og programpilen hopper til linie 000.

Bemærk: [SST] kan ikke tvinges ind i tomme programlinier. Så hvis den sidste instruktion i et program ikke er [GTO] eller [GSB], vil programpilen automatisk hoppe op til linie 000. Er den sidste instruktion derimod [GTO] eller [GSB], hopper programpilen hen til det programafsnit, hvis indledende etiket er specificeret af hopordrens adresse.

Brug af [SST] og [g] [BST] i programmeringsstatus

Gennemsyn af det lagrede program. Hvis man i programmeringsstatus trykker på [SST] og slipper tasten med det samme, vil programpilen rykke frem til næste linie, hvis indhold udlæses. [g] [BST] virker på samme måde, bortset fra at programpilen rykker en linie tilbage.

Indtastning	Lyspanel	
g P/R	*000-*	Skift til programmerings-status.
SST	*001- 42, 21, 15*	
SST	*002- 43 11*	
SST	*003- 34*	
SST	*004- 43 11*	
SST	*005- 40*	
SST	*006- 11*	
SST	*007- 43 32*	
g P/R		Skift til normalstatus.

Hurtigt gennemsyn af program. Hvis man holder SST eller BST nedtrykket i programmeringsstatus, vil programpilen bevæge sig frem eller tilbage, lige så længe som tasten holdes nede. Den første programlinie udlæses i ca. 2 sekunder og de følgende i ca. halvandet sekund, inden programpilen går videre til næste/foregående linie.

Modificering af et program

For at illustrere, hvordan HP-11C's editeringsfunktioner virker, vil vi modificere trekantsprogrammet således, at programmet automatisk udlæser indholdet af X-registret efter visse operationer. Dette kan gøres ved at bruge f PSE *(pause)*.

Indtastning	Lyspanel	
f LBL E	*001- 42, 21, 15*	
g x^2	*002- 43 11*	– Der skal indsættes en
$x \gtrless y$	*003- 34*	f PSE -instruktion
g x^2	*004- 43 11*	– efter hver af disse tre
+	*005- 40*	– linier.
$\sqrt{x}$	*006- 11*	
g RTN	*007- 43 32*	

Indsættelse af instruktioner. Der kan indsættes nye programlinier overalt i programhukommelsen:

1. Skift til programmeringsstatus med g P/R
2. Brug SST eller GTO · *nnn* til at placere programpilen ved programlinien *lige før* det sted, hvor den nye instruktion skal være.
3. Indtast den nye instruktion, som bliver lagret i *næste* programlinie.

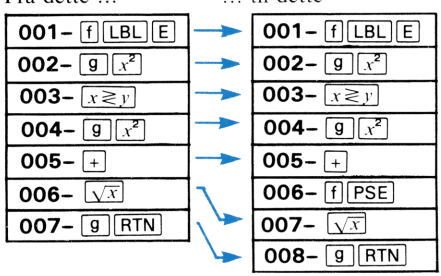
Alle programlinier efter denne skubbes en linie frem i programhukommelsen.

Da hver eneste ny instruktion skubber alle de følgende en linie frem, er det lettest at starte med den linie, der har det højeste linienummer, hvis man skal udføre flere rettelser. På den måde kan man hele tiden bruge de oprindelige linienumre til at finde frem til de steder i programmet, hvor noget skal ændres.

Eksempel. Der indsættes en [f] [PSE] -instruktion efter [+]:

Indtastning	**Lyspanel**	
[GTO] [·] 000		Sæt programpilen til linie 000.
[g] [P/R]	***000-***	Linie 000.
[SST]	***001-42, 21, 15***	Brug [SST] til at nå
[SST]	***002- 43 11***	frem til linie 005.
[SST]	***003- 34***	
[SST]	***004- 43 11***	
[SST]	***005- 40***	Linie 005.
[f] [PSE]	***006- 42 31***	Linie 006 indeholder nu pauseinstruktionen.

Med programpilen ud for linie 005 blev programhukommelsen ændret således, da De trykkede [f] [PSE]:



Hop til et bestemt linienummer. Det er klart, at det kan tage urimeligt lang tid at nå frem til en linie langt fremme i hukommelsen, hvis man alene er henvist til at bruge [SST]. Men med [GTO] [·] *nnn* kan man undgå spild af tid. Når man bruger [GTO] [·] *nnn*, hopper programpilen straks hen til linie *nnn*. [GTO] [·] *nnn* kan bruges i såvel normal- som programme-

ringsstatus, uden at der udføres nogen programinstruktioner – [GTO] [·] *nnn* flytter kun programpilen til linie *nnn*.
Hvis man f.eks. vil indsætte en [f] [PSE] efter den anden [g] [x^2] -instruktion, der er lagret i linie 004, kan det gøres således:

Indtastning		Lyspanel		
[GTO] [·]	004	***004-***	***43 11***	Hop til linie 004.
[f] [PSE]		***005-***	***42 31***	Indsættelse af [f] [PSE] efter linie 004.

Da [f] [PSE] blev lagret efter linie 004, ændredes programhukommelsen.

Fra dette ... ... til dette.

001- [f] [LBL] [E]	001- [f] [LBL] [E]
002- [g] [x^2]	002- [g] [x^2]
003- [$x \geqslant y$]	003- [$x \geqslant y$]
004- [g] [x^2]	004- [g] [x^2]
005- [+]	005- [f] [PSE] ← [f] [PSE] blev indsat her. Alle følgende instruktioner i programhukommelsen, skubbes én linie frem.
006- [f] [PSE]	006- [+]
007- [$\sqrt{x}$]	007- [f] [PSE]
008- [g] [RTN]	008- [$\sqrt{x}$]
	009- [g] [RTN]

Vi vil nu bruge [g] [BST] til at sætte den sidste [f] [PSE] ind i programmet.

Indtastning	Lyspanel		
	005-	***42 31***	Programpilens placering efter den sidste modifikation.
[g] [BST]	***004-***	***43 11***	
[g] [BST]	***003-***	***34***	
[g] [BST]	***002-***	***43 11***	Indsæt den sidste [PSE] efter denne linie.
[f] [PSE]	***003-***	***42 31***	[f] [PSE] indsættes her. Alle følgende instruktioner i programhukommelsen skubbes én linie frem.

Nu er modifikationerne udført. De kan eventuelt bruge [SST] til at kontrollere, at det lagrede program er det samme som vist herunder.

Indtastning	Lyspanel	
[f] [LBL] [E]	*001-*	*42, 21, 15*
[g] [x^2]	*002-*	*43 11*
[f] [PSE]	*003-*	*42 31*
[$x \geqslant y$]	*004-*	*34*
[g] [x^2]	*005-*	*43 11*
[f] [PSE]	*006-*	*42 31*
[+]	*007-*	*40*
[f] [PSE]	*008*	*42 31*
[$\sqrt{x}$]	*009-*	*11*
[g] [RTN]	*010-*	*43 32*

Vi vil nu prøve at køre det nye program og bruger igen værdierne 22 og 9 for a og b.

Indtastning	Lyspanel	
[g] [P/R]		Skift til normaltilstand.
22 [ENTER]	*22,0000*	Indtastning af a (Y).
9	*9*	Indtastning af b (X).
[f] [E]	*81,0000*	Pause, hvor b^2 udlæses.
	484,0000	Pause, hvor a^2 udlæses.
	565,0000	Pause, hvor $a^2 + b^2$ udlæses.
	23,7697	Slutresultatet, c, udlæses.

Forsøg på indsættelse af instruktioner i fuld programhukommelse

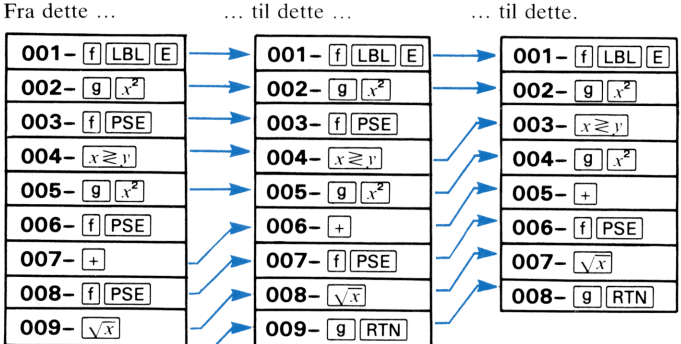
Hvis der er lagret instruktioner i alle 203 programlinier, kan der ikke indsættes nye instruktioner. Forsøger man alligevel, reagerer HP-11C med fejlmeddelelsen ***Error 4,*** uden at der sker mere med hukommelsen.

Sletning af instruktioner. Under en editeringsproces sker det ofte, at man skal have slettet en linie. Det gøres ved at bruge [GTO] . [9] [BST] eller [SST] til at sætte programpilen ud for den linie, der skal slettes. Med HP-11C i programmeringsstatus kan man derefter slette linien med [←]. Så snart linien er slettet, rykker alle de følgende instruktioner én linie op. Endelig udlæses linien *før* den, der blev slettet.

Eksempel. Slet to pauser i trekantsprogrammet, så kun pausen efter kvadratsummen bevares. De skal altså slette instruktionerne i linie 003 og linie 006. Husk, hvad der tidligere blev skrevet om at starte fra det højeste linienummer.

Indtastning	Lyspanel		
[g] [P/R]	***000-***		Skift til programmeringsstatus.
[GTO] [·] 006	***006-***	***42 31***	Hop til linie 006.
[←]	***005-***	***43 11***	Sletter [f] [PSE] og udlæser den foregående linie (005). De følgende instruktioner rykker én linie op.
[GTO] [·] 003	***003-***	***42 31***	Hop til linie 003.
[←]	***002-***	***43 11***	Sletter [f] [PSE] og udlæser den foregående linie.

Da De slettede pauseinstruktionerne i linierne 003 og 006, ændredes programhukommelsen.



Hvis De har modificeret trekantsprogrammet som beskrevet ovenfor, indeholder programmet nu kun én pause, i hvilken kvadratsummen udlæses. Derefter beregnes længden af hypotenusen, og programmet stopper med dette resultat i X-registret.

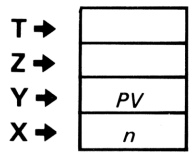
Kør programmet med a og b som 17 og 34 meter.

Indtastning	Lyspanel	
[g] [P/R]		Skift til normalstatus.
17 [ENTER] 34	***34***	
[f] [E]	***1.445,0000***	Kvadratsummen udlæses i ca. 1 sekund.
	38,0132	Længden af hypotenusen.

Når man sletter instruktioner i et program på mere end 63 linier, vil den automatiske splitning sørge for konvertering fra programregistre til dataregistre. Hvis man f.eks. sletter en eller anden linie i et program på 71 linier, vil HP-11C automatisk konvertere linierne 71-77 til dataregistret $R_{.8}$.

Opgaver

Det følgende program kan bruges til at beregne slutværdien af et beløb, der sættes til forrentning. Programmet bygger på formlen $FV = PV\ (1+i)^n$, hvor FV er slutværdien, PV er det oprindelige beløb, i er rentesatsen, og n er antallet af beregningsterminer.



Det antages, at PV og n indtastes som vist på tegningen til højre, og at i er 7,5% pr. termin.

Indtastning	Lyspanel	
[f] [LBL] [B]	***001- 42, 21, 12***	
[f] [FIX] 2	***002- 42, 7, 2***	
1	***003- 1***	
[.]	***004- 48***	
0	***005- 0***	Rentesats
7	***006- 7***	
5	***007- 5***	
[x≷y]	***008- 34***	
[y^x]	***009- 14***	$(1+i)^n$
[×]	***010- 20***	$PV(1+i)^n$
[g] [RTN]	***011- 43 32***	

a. Indtast programmet.
b. Brug programmet til at bestemme slutværdien af en opsparing på 1.000 kr. over 5 terminer.
(Svar: 1.435,63 kr.)
Find dernæst slutværdien for 2.300 kr., der står i 4 terminer.
(Svar: 3.071,58 kr.)
c. Ret programmet, så rentesatsen bliver 8% i stedet for 7,5%.
d. Kør programmet med den nye rentesats og find slutværdien af 500 kr., der trækker rente i 4 terminer og af 2.000 kr., der står i 10 terminer.
(Svar: 680,24 kr.; 4.317,85 kr.)

2. Følgende program beregner den tid det tager for en genstand at ramme jorden, når den slippes i en given højde. (Der ses bort fra luftmodstand og lignende). Når højden h (meter) indtastes, udlæses tiden t (sek) med [f] [B]. Programmet bygges op over formlen:

$$t = \sqrt{\frac{2h}{9.8}}$$

a. Slet programhukommelsen, vælg fastkomma 4 og indtast nedenstående program.

Indtastning	Lyspanel	
[f] CLEAR [PRGM]	*000-*	
[f] [LBL] [B]	*001-*	*42, 21, 12*
2	*002-*	*2*
[×]	*003-*	*20*
9	*004-*	*9*
[.]	*005-*	*48*
8	*006-*	*8*
[÷]	*007-*	*10*
[√x]	*008-*	*11*
[g] [RTN]	*009-*	*43 32*

b. Brug programmet til at beregne, hvor lang tid en sten er om at falde fra toppen af Eifeltårnet (300,51 meter) og fra en luftballon, der befinder sig 1.000 meter over jordens overflade.
(Svar: 7,8313 sek; 14,2857 sek.)
c. Lav programmet om, så det beregner tiden, når højden er givet i *feet*.

Hertil bruges formlen:

$$t = \sqrt{\frac{2h}{32.1740}}$$

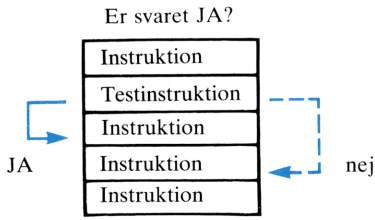
d. Brug det modificerede program til at bestemme, hvor lang tid en sten vil være om at falde ned fra toppen af Grand Coulee dæmningen (550 feet) og fra den 1350 feet høje World Trade Center bygning i New York.
(Svar: 5,8471 sek; 9,1607 sek.)

Afsnit 3

Forgreninger og kontroloperationer

Betingede instruktioner

En betinget instruktion er en instruktion, hvis funktion er afhængig af udfaldet af en sammenligning. Den har form af et spørgsmål, og man taler derfor om *testinstruktionen* og dens *udfald,* som kan være *JA* (sand) eller *nej* (falsk). Hvis svaret på spørgsmålet er JA, udføres instruktionen i næste linie, men er svaret nej, overspringes næste linie.



Instruktionen efter testinstruktionen udføres, hvis svaret er JA

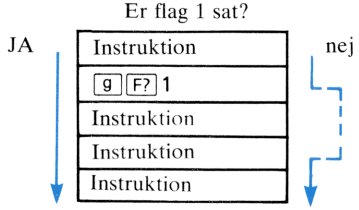
HP-11C's betingede instruktioner sammenligner (tester) indholdet af X- og Y-registrene – dvs. tallene x og y. Besvares spørgsmålet med JA, bliver instruktionen lige efter testinstruktionen udført, ellers bliver den sprunget over.

Test	Spørgsmål
f $x \leq y$	Er x mindre end eller lig med y?
f $x > y$	Er x større end y?
f $x \neq y$	Er x og y forskellige?
f $x = y$	Er x og y eksakt identiske?
g $x < 0$	Er x negativ?
g $x > 0$	Er x positiv og forskellig fra 0?
g $x \neq 0$	Er x forskellig fra 0?
g $x = 0$	Er x eksakt lig 0?

Flag

Der er en anden type betingede instruktioner, hvor man ikke sammenligner indholdet af to registre, men spørger om en bestemt status. Det gøres ved hjælp af flag. Et flag kan være *sat* (sand) eller *slettet* (falsk). Et program kan teste et flag og derved afgøre, hvilket af to videre forløb der skal vælges.

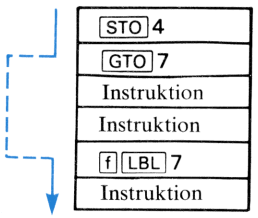
HP-11C har to flag, flag 0 og flag 1. Et flag *sættes* ved at trykke [g] [SF] efterfulgt af 0 eller 1. Tilsvarende kan et flag *slettes* med [g] [CF] 0 eller 1. *Testinstruktionen* er [g] [F?] 0 eller 1, hvor næste instruktion udføres, hvis flaget er sat (JA/sand), hvorimod den springes over, hvis flaget er slettet (nej/falsk).



Et flag, der er sat med [g] [SF] *n*, forbliver sat, indtil det slettes af [g] [CF] *n*, eller til hele hukommelsen slettes.

Ubetinget hop

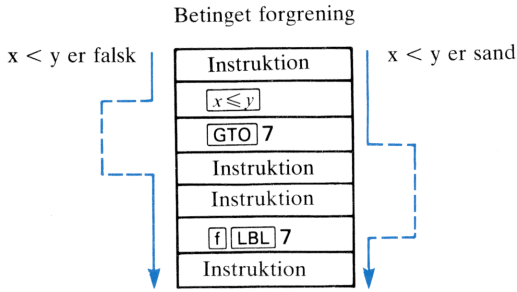
Go To. De har allerede brugt [GTO] til at flytte programpilen hen til en bestemt linie i programmet i forbindelse med editering. Men den kan også bruges under programkontrol til at flytte programpilen hen til en specificeret etiket. Dette *hop* udføres ganske uafhængigt af, hvilken status maskinen er i og af, hvilke tal der står i X- og Y-registrene; der er altså tale om et *ubetinget* hop. Når [GTO] bruges som hopinstruktion, skal den efterfølges af en adresse: en alfabetisk eller en numerisk etiket.



Når programmet møder [GTO] *etiket*, stopper programudførelsen, medens programpilen søger frem i programhukommelsen, indtil den møder *etiketten*. Så snart det sker, fortsætter programmet med at udføre de instruktioner, der er lagret efter etiketten.

Forgrening og løkker

De betingede instruktioner kan kobles sammen med [GTO] således, at [GTO] udføres, hvis resultatet af testen er JA. Man opnår altså en konstruktion, hvor det er resultatet af den foregående testinstruktion, der afgør om [GTO] skal udføres. Konstruktionen, der kaldes en *betinget forgrening*, karakteriseres ved, at man hopper, hvis svaret er JA.



Hop hvis JA

Når HP-11C er i normalstatus, vil [GTO] *etiket* føre programpilen hen til *etiket* uden at starte programmet. Det giver en bekvem mulighed for at gennemse eller editere de linier, der følger efter en etiket.

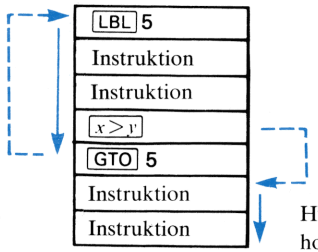
Løkker. Løkker er specielle betingede forgreninger, hvor [GTO] bruges til at gentage et programafsnit vilkårligt antal gange. En løkke kan indeholde en tæller, der bestemmer, hvor mange gange programafsnittet skal udføres, eller man kan lade en testinstruktion være bestemmende for, hvornår programafsnittet ikke længere skal gentages. Under alle omstændigheder indeholder løkken et sæt instruktioner, der gør det muligt dels at gentage løkken, dels at hoppe ud af den.

I nedenstående eksempel vil løkken blive gentaget, indtil resultatet af testen lige før [GTO] 5 er falsk (nej). Så længe betingelsen er opfyldt, vil [GTO] 5 blive udført.

Man skal altid overveje sine testinstruktioner nøje: hvis testen viser sig altid at være sand, får man herved skabt en *uendelig løkke,* hvorved programmet vil fortsætte med at udføre de samme instruktioner igen og igen, indtil det stoppes manuelt.

Betinget forgrening

Kontrol af løkke



Hvis testen er sand, fortsættes i løkken.

Hvis testen er falsk, hoppes ud af løkken.

Betinget forgrening med løkkestruktur. Det følgende program beregner og udlæser kvadratroden af heltallene 1 til 10. Når kvadratroden af et tal er beregnet og udlæst, bruges [f] [x>y] til at afgøre, om tallet var mindre end 10. Hvis tallet faktisk var mindre end 10 (testen er sand), bruges [GTO] 0 til at gentage beregningen for det næste heltal. Hvis testen derimod er falsk, overspringes hopinstruktionen, og programkørslen afsluttes.

Indtastning	**Lyspanel**	
[g] [P/R]	***000-***	Skift til programmeringsstatus.
[f] CLEAR [PRGM]	***000-***	Slet programhukommelsen.
[f] [LBL] [C]	***001- 42, 21, 13***	
0	***002- 0***	Nulstiller R_1, der bruges
[STO] 1	***003- 44 1***	som tæller i løkken.
[f] [LBL] 0	***004- 42, 21, 0***	Løkken starter her.

Indtastning	Lyspanel	
1	*005- 1*	
STO + 1	*006- 44, 40, 1*	Heltalstæller.
RCL 1	*007- 45 1*	
f PSE	*008- 42 31*	Udlæser næste heltal.
$\sqrt{x}$	*009- 11*	Kvadratroden af tallet.
f PSE	*010- 42 31*	Udlæser kvadratroden.
g LSTx	*011- 43 36*	Henter SIDSTE X.
1	*012- 1*	
0	*013- 0*	Sammenligner sidste X-
f $x>y$	*014- 42 20*	værdi med 10.
GTO 0	*015- 22 0*	Betinget forgrening. Hvis sidste X-værdi er mindre end 10, fortsættes i løkken ...
g CLx	*016- 43 35*	... ellers afsluttes pro-
g RTN	*017- 43 32*	grammet.
g P/R		Skift til normalstatus.

Programmet startes med f C . Lommeregneren udlæser en tabel over de 10 første heltal og deres kvadratrødder. Programudførelsen er færdig, når 0,0000 udlæses.

Programforløbet. Når man trykker f C , søger programpilen gennem programhukommelsen, indtil den finder f LBL C . De følgende instruktioner til og med linie 014 f $x>y$ udføres i den lagrede rækkefølge.
Hvis resultatet af testen er JA (dvs. at det sidste tal var mindre end 10), bliver den følgende GTO 0 udført, og programmet kører løkken igennem igen. Hvis udfaldet af testen er »falsk«, overspringes hopinstruktionen, og i stedet udføres g CLx i linie 016, hvorefter programmet stopper.
Kapitel 3, afsnit 2 indeholder bl.a. eksempler på brug af løkker.

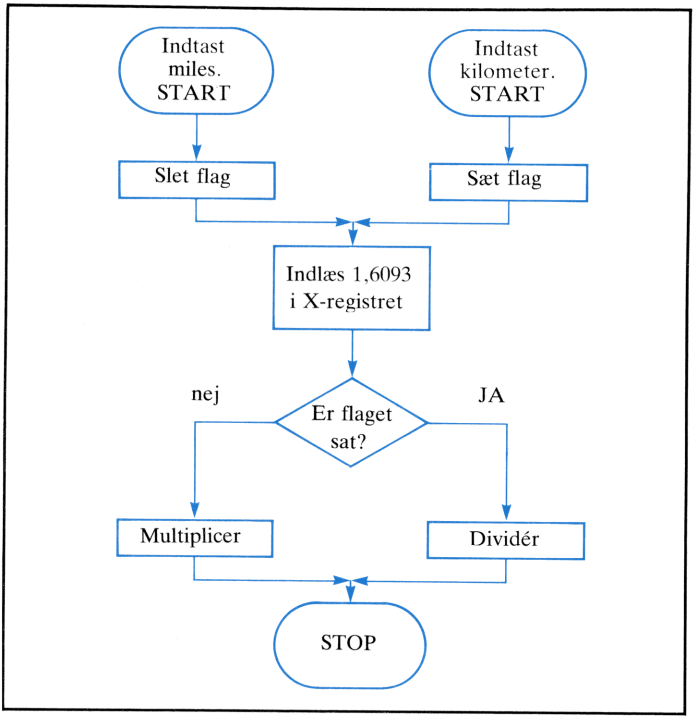
Opgave: Skriv et program, der beregner og udlæser salgsprovisioner. For salg under 1.000 kr. udgør provisionen 8%, for salg over 1.000 kr. udgør den 7%. Brug en betinget forgrening til at løse opgaven.

Brug af flag

Testinstruktionerne, hvor man sammenligner x med y eller 0 ($x?y$ og $x?0$), giver mulighed for at lade det hidtidige programforløb afgøre, om en

instruktion skal udføres eller springes over. Flagene har samme effekt, men en anden baggrund, idet de bruges til at markere en status. Hvis man f.eks. sætter flag 1, når et bestemt programafsnit udføres, vil man senere kunne afgøre, om netop dette afsnit er blevet udført.

Eksempel: Følgende program omregner mellem miles og kilometer. [A] bruges til omregning fra miles til kilometer, hvorimod [B] omregner fra kilometer til miles. Et sådant program kunne have været lavet ret enkelt: under [f] [LBL] [A] ganges miles med 1,6093, og under [f] [LBL] [B] divideres kilometer med 1,6093, og 1,6903 lagres som programinstruktioner eller i et dataregister. Vi vil bruge en mere raffineret fremgangsmåde, hvor man udnytter, at samme omregningsfaktor bruges i de to tilfælde. Det skal dog medgives, at eksemplet først og fremmest illustrerer brugen af flag.



Indtastning	Lyspanel	
[g] [P/R]	*000-*	Skift til programmerings-status.
[f] CLEAR [PRGM]	*000-*	Slet program-hukommelsen.
[f] [LBL] [A]	*001- 42, 21, 11*	Program, der omregner miles til kilometer.
[g] [CF] 0	*002- 43, 5, 0*	Flag 0 slettes (som tegn på, at inddata er givet i *miles*).
[GTO] 1	*003- 22 1*	Ubetinget hop til [LBL] 1.
[f] [LBL] [B]	*004- 42, 21, 12*	Program, der omregner kilometer til miles.
[g] [SF] 0	*005- 43, 4, 0*	Flag 0 sættes (som tegn på, at inddata er givet i kilometer).
[f] [LBL] 1	*006- 42, 21, 1*	Etiket, der markerer det programafsnit, hvor selve omregningen udføres.
1	*007- 1*	Omregningsfaktoren lagres i X-registret.
[.]	*008- 48*	
6	*009- 6*	
0	*010- 0*	
9	*011- 9*	
3	*012- 3*	
[g] [F?] 0	*013- 43, 6, 0*	Flagtest. Hvis JA (km
[GTO] 2	*014- 22 2*	til miles), hoppes til etiket »2«.
[×]	*015- 20*	Hvis flagtest er falsk
[g] [RTN]	*016- 43 32*	(miles til km), ganges med omregningsfaktoren, og resultatet udlæses.
[f] [LBL] 2	*017- 42, 21, 2*	Omregner km til
[÷]	*018- 10*	miles, og udlæser
[g] [RTN]	*019- 43 32*	resultatet.
[g] [P/R]		Skift til normalstatus.

Brug programmet til at omregne 26 miles til km og bagefter til at omregne 1,5 km til miles.

Indtastning	Lyspanel	
26	***26***	Miles.
f A	***41,8418***	26 miles omregnet til km.
1.5	***1,5***	Kilometer.
f B	***0,9321***	1,5 km omregnet til miles.

Programforløbet: Der indtastes et tal, og A og B bruges til at angive, hvilken omregning der skal finde sted. Programmet sætter eller sletter flag 0 og udfører derefter hovedprogrammet (under etiket »1«). I dette tilfælde består hovedprogrammet af indlæsning af omregningsfaktoren og en flagtest, hvor der spørges, om flag 0 er sat. Hvis flaget er sat (JA/sand), udføres næste instruktion. Hvis flaget ikke er sat (nej/falsk), springes næste instruktion over. Instruktionen efter flagtesten er GTO 2, der fører programpilen ned til et programafsnit, hvor tallet i Y- divideres med tallet i X-registret, og resultatet udlæses. Hvis flaget er slettet, springes hoporden over, og tallene i X- og Y-registrene ganges, og resultatet udlæses. Bemærk, at tallene i X- og Y-registrene ikke berøres af den betingede forgrening, og at flaget bruges til at opbevare information, som bruges senere i programmet.

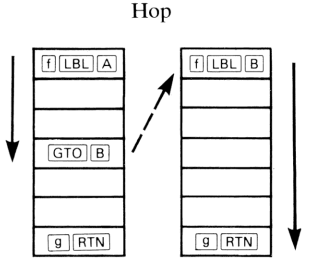
Afsnit 4

Underprogrammer

Det sker tit, at et program indeholder en række instruktioner, der skal bruges flere steder i programmet. Når det sker, kan man normalt spare programplads ved at lade programafsnittet optræde som et *underprogram,* dvs. et programafsnit, der omsluttes af [f] [LBL] *etiket* og [g] [RTN], og som virker som en *brugerdefineret* funktion. Det betyder, at underprogrammet, kan bruges (kaldes) overalt i programmet.

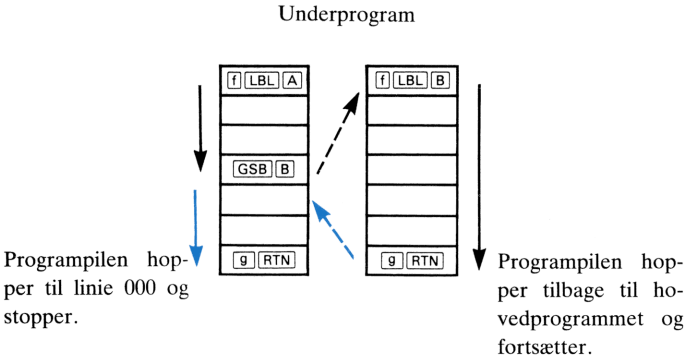
Kald af underprogram

Et underprogram *kaldes* ved hjælp af [GSB] efterfulgt af en alfa- eller numerisk etiket*. [GSB] fører programpilen hen til den angivne etiket – ligesom [GTO]. Men der er den forskel, at når programmet møder [g] [RTN], hopper programpilen hen til instruktionen *lige efter* [GSB]-instruktionen, hvorfra programudførelsen fortsætter. Kald af underprogram kaldes også *forgrening med returhop.* Den del af programmet, der kalder underprogrammet, betegnes ofte som *hovedprogrammet.*



Programpilen hopper til linie 000 og stopper.

*) Der kan bruges forkortet indtastning, når [GSB] bruges med en alfaetiket.

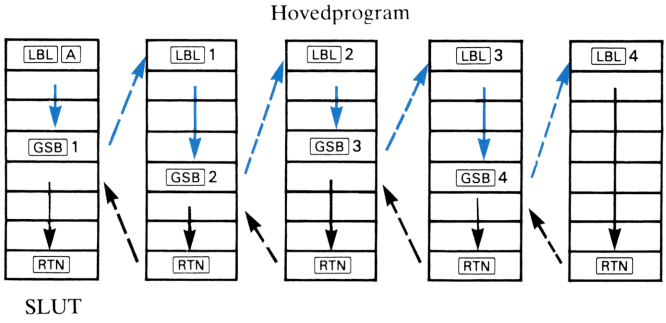


Som De kan se, ligger den eneste forskel på GTO og GSB i returhoppet.

Efter GTO vil den næste g RTN føre programpilen tilbage til linie 000, hvor programmet stoppes.
Efter GSB vil den næste g RTN føre programpilen tilbage til linien umiddelbart efter GSB, hvorfra programmet kører videre.

Underprogrammer i flere niveauer

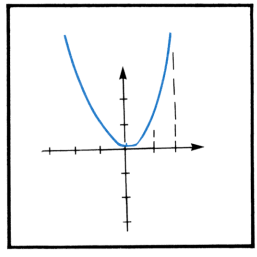
Et underprogram kan kalde et andet underprogram, som igen kan kalde et underprogram osv. Styringen af dette foregår ved hjælp af et særligt sæt registre, der formelt burde hedde underprogramreturadresseregisterstakken, men som vi her vil kalde returstakken. Hver gang man kalder et underprogram, lagres linienummeret for linien lige efter GSB i det nederste af registrene i returstakken, som samtidig løftes. Der er plads til fire returadresser i stakken, så man kan altså have et hovedprogram og fire underprogrammer, hvoraf de tre desuden fungerer som hovedprogrammer.



Hvis man prøver at lægge en femte returadresse i returstakken, stopper programmet, og HP-11C udlæser fejlmeddelelsen ***Error 5.*** Bemærk, at når der udføres en [9] [RTN], vil HP-11C altid undersøge, om der står en ventende returadresse i returstakken. Hvis det er tilfældet, vil den blive brugt som adresse for programpilen. Derefter falder returstakken, men i modsætning til rullestakkens T-register er returstakkens øverste register ikke selvkopierende. Det betyder, at når alle ventende returhop er udført, vil alle returadresser være sat til nul. Derfor vil [9] [RTN] føre programpilen til linie 000, hvis instruktionen ikke mødes som afslutning på et underprogram.

Brug af underprogrammer

Eksempel: Skriv et program, der beregner den gennemsnitlige hældningskoefficient i intervallet x_1 til x_2 for den viste kurve: $y = x^2 - \sin x$.



Løsning: Middelværdien for hældningskoefficienten er givet ved:

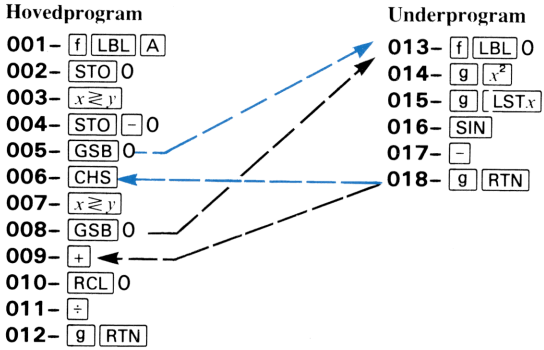
$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{(x_2^2 - \sin x_2) - (x_1^2 - \sin x_1)}{x_2 - x_1}$$

Bemærk, at x^2 – sin x skal beregnes *to* gange: én gang for x_1 og én gang for x_2.

Da løsningen indebærer, at den samme beregning skal foretages to gange undervejs, kan vi lave et underprogram, der sparer plads i programhukommelsen.

(Det forudsættes, at grader – [DEG] – er valgt som vinkelmål).

Med x_1 i Y-registret og x_2 i Y-registret vil [f] [A] starte programmet fra [f] [LBL] [A] i linie 001. Når programmet møder [GSB] 0, hopper programpilen hen til linie 013 og udfører instruktionerne 013-017. [g] [RTN] i linie 018 bringer programpilen tilbage til linien lige efter underprogramkaldet: linie 006. Tilsvarende sker ved kaldet i linie 008.



Hvis man f.eks. sætter x_1 til 2 og x_2 til 3, vil beregningen forløbe således:

	001	002	003	004
T →				
Z →				
Y →	2	2	3	3
X →	3	3	2	2

Keys →	[f] [LBL] [A]	[STO] 0	[x≷y]	[STO] [−] 0
	x_1 i Y x_2 i X	x_2 i R_0	Ombytning	x_2-x_1 i R_0

	005	013	014	015
T →				
Z →				3
Y →	3	3	3	4
X →	2	2	4	2
Indtastning	GSB 0	f LBL 0	g x^2	g LST x
	Hop til etiket 0	Start på underprogram	x_1^2	x_1

	016	017	018	006
T →				
Z →	3			
Y →	4	3	3	3
X →	0.0349	3.9651	3.9651	-3.9651
Indtastning	SIN	−	g RTN	CHS
	sin x_1	x_1^2 -sin x_1 = y_1	Retur til hoved program	-(x_1^2-sin x_1 -sin x_1)

Fra linie 019 hopper programpilen tilbage til hovedprogrammet og fortsætter fra linien umiddelbart efter den GSB-instruktion, der blev brugt til at kalde underprogrammet. Når GSB 0 igen mødes i linie 008, hopper programpilen igen til linie 013:

	007	008	013	014
T →				
Z →				
Y →	-3.9651	-3.9651	-3.9651	-3.9651
X →	3	3	3	9
Indtastning	$x \gtrless y$	GSB 0	f LBL 0	g x^2
	Ombytning	Hop til etiket 0	Start på underprogram	x_2^2

	015	016	017	018
T →				
Z →	-3.9651	-3.9651		
Y →	9	9	-3.9651	-3.9651
X →	3	0.0523	8.9477	8.9477
Indtastning	g LSTx	SIN	−	g RTN
	x_2	sin x_2	x_2^2-*sin* x_2 = y_2	Retur til hovedprogram

Når underprogrammet er udført for anden gang, hopper programpilen tilbage til linie 009, linien umiddelbart efter underprogramkaldet. Nu er y_2 i X-registret, medens $-y_1$ er i Y-registret. De efterfølgende instruktioner afslutter beregningen.

	009	010	011	012
T →				
Z →				
Y →		4.9826		
X →	4.9826	1	4.9826	4.9826
Indtastning	+	RCL 0	÷	g RTN
	y_2-y_1	x_2-x_1	$\frac{y_2-y_1}{x_2-x_1}$	SLUT

Når programmet stopper, udlæses hældningskoefficienten for den rette linie over intervallet x_1 og x_2.

Prøv nu selv at køre programmet for følgende værdier af x_1 og x_2:

0,5 og 1,25; 2,52 og 3,72; 5 og 7.

(Svar: 1,7325; 6,2226; 11,9826.)

Kapitel 3, afsnit 2 indeholder bl.a. nærmere anvisninger på brugen af underprogrammer.

Bemærk, at [GTO] og [GSB] altid søger *frem* i programhukommelsen. Det giver mulighed for at bruge den samme etiket mere end én gang. Det kan være en farlig metode, som kan give anledning til de mest komplicerende logiske fejl i et program, men under visse omstændigheder er det et såvel legitimt som fornuftigt programmeringsmiddel.

Eksempel: Følgende program bestemmer 2-normen af tallene *x*, *y*, *z* og *t*, dvs. kvadratroden af tallenes kvadratsum. [f] [LBL] [A] vil blive brugt som etiket for såvel hovedprogram som underprogram. Programmet udføres ved at lagre *x*, *y*, *z* og *t* i henholdsvis X, Y, Z og T, hvorefter [f] [LBL] [A] trykkes.

Indtastning	**Lyspanel**	
[g] [P/R]	***000-***	
[f] CLEAR [PRGM]	***000-***	
[f] [LBL] [A]	***001- 42, 21, 11***	
[g] [x^2]	***002- 43 11***	x^2.
[GSB] [A]	***003- 32 11***	y^2 og x^2+y^2.
[GSB] [A]	***004- 32 11***	z^2 og $x^2+y^2+z^2$.
[GSB] [A]	***005- 32 11***	t^2 og $x^2+y^2+z^2+t^2$.
[$\sqrt{x}$]	***006- 11***	$\sqrt{x^2+y^2+z^2}$.
[g] [RTN]	***007- 43 32***	
[f] [LBL] [A]	***008- 42, 21, 11***	
[$x \geqslant y$]	***009- 34***	
[g] [x^2]	***010- 43 11***	
[+]	***011- 40***	
[g] [RTN]	***012- 43 32***	
[g] [P/R]		

Prøv programmet med følgende værdier:

$x = 4{,}3 \quad y = 7{,}9 \quad z = 1{,}3 \quad t = 8{,}0$

Indtastning	**Lyspanel**
8 [ENTER]	***8,0000***
1.3 [ENTER]	***1,3000***
7.9 [ENTER] 4,3	***4,3***
[f] [A]	***12,1074***

Afsnit 5

Indeksregistret

Indeksregistret (R_I) er et af HP-11C's allervigtigste programmeringsredskaber. Det er et permanent dataregister, og kan som sådan bruges til lagring af tal – men desuden kan det bruges som:

- Tælleregister i forbindelse med kontrol af løkker.
- Adresseregister ved indirekte adressering af dataregistre, forgreninger og underprogrammer.

Indeksregistret bruges ved, at man lagrer et kontroltal i indeksregistret. *Heltals*delsen af dette kontroltal bestemmer, hvor mange gange en løkke skal udføres, eller hvilken adresse der skal hoppes til. *Decimal*delen består af parametre, der bruges til at ændre og angive grænser for heltalsdelen.

[I] og [(i)] med forkortet indtastning. De behøver ikke bruge [f] i forbindelse med [I] og [(i)]. Hvis De alligevel bruger [f], registreres den ikke i programhukommelsen – så De kan altså roligt bruge den, hvis De helst vil det.

Indeksregistrets direkte funktioner

Indeksregistrets direkte funktioner virker på selve tallet i indeksregistret.

Lagring og fremkald af tallet i indeksregistret. [STO] [I] og [RCL] [I] virker på præcis samme måde som de sædvanlige [STO] - og [RCL]-operationer.

Ombytning af X og I. [f] [$x \gtrless$ I] ombytter indeholdene af X-registret og indeksregistret (r_I eller I-registret) på samme måde, som [$x \gtrless y$] ombytter tallene i X- og Y-registrene.

Udførelse af	Eksempel	Lyspanel
Lagring i R_I	12345	***12.345***
	STO I	***12.345,0000***
(Nulstil lyspanel)	←	***0,0000***
Fremkald tallet i R_I	RCL I	***12.345,0000***
(Nulstil lyspanel)	←	***0,0000***
Ombyt X og I	f $x \geq I$	***12.345,0000***

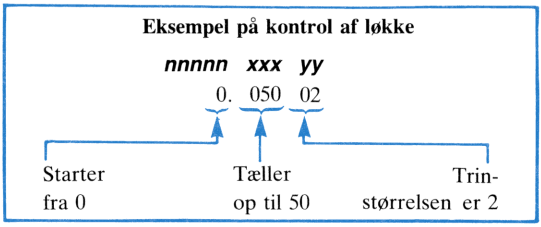
Inkrementering og dekrementering af indeksregistret. f ISG *(inkrementér og skip, hvis større end)* og f DSE *(dekrementér og skip, hvis mindre end eller lig med)* er avancerede løkkefunktioner, der bruger kontroltallet i indeksregistret til at afgøre, hvor mange gange en løkke skal udføres, og hvordan løkketælleren skal arbejde.
Kontroltallets heltalsdel kan desuden bruges til indirekte adressering af dataregistre, forgreninger og kald af underprogrammer.
Både f ISG og f DSE fortolker og sammenligner kontroltallets forskellige komponenter efter følgende formatregler:

	+nnnnn	er den aktuelle tællerværdi
nnnnn.xxxyy hvor	***xxx***	er den aktuelle testværdi
	yy	er inkrement/dekrementværdien; dvs. det tal, der ved hver brug af funktionen lægges til/trækkes fra tællerværdien

nnnnn er kontroltallets heltalsdel og bruges som tæller og som adresseangivelse ved indirekte adressering af dataregistre, forgreninger og kald af underprogrammer. ***nnnnn*** kan have fra 1 til 5 cifre og vil antage værdien 0, hvis ikke andet specificeres. ***nnnnn*** inkrementeres af f ISG og dekrementeres af f DSE som angivet af ***yy.***

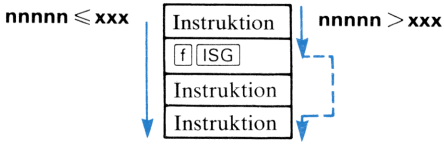
xxx er de første tre cifre i kontroltallets decimaldel. Når f ISG eller f DSE bruges til at inkrementere eller dekrementere ***nnnnn,*** bliver ***xxx*** internt sammenlignet med ***nnnnn*** for derved at afgøre, om løkken er udført det ønskede antal gange. ***xxx*** *skal* opgives som et 3-cifret tal, dvs. at en ***xxx***-værdi på 10 skal opgives som 010. (***xxx*** er en referenceværdi og ændres som sådan ikke af f ISG og f DSE.

yy er 4. og 5. ciffer i kontroltallets decimaldel. ***yy*** bestemmer, hvor store inkrementer og dekrementer der skal bruges – dvs. hvor meget der hver gang skal lægges til eller trækkes fra ***nnnnn. yy*** *skal* altid opgives som et 2-cifret tal (f.eks. 02, 02 eller 55) og vil antage værdien 01, hvis ikke andet specificeres. (***yy*** er en referenceværdi og ændres som sådan ikke af [f] [ISG] og [f] [DSE]).

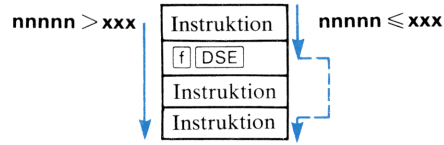


Brug af [f] [ISG] og [f] [DSE]. Inkrement- og dekrementfunktionerne bruger ***nnnnn/xxx*** på samme måde som de tidligere omtalte testinstruktioner.

Ved hver udførelse af [f] [ISG] lægges ***yy*** til ***nnnnn.*** Derefter undersøger programmet, om ***nnnnn*** er større end ***xxx***; hvis det er tilfældet, springes næste linie over (den *skippes*), og programmet fortsætter kørslen med den følgende instruktion.



Ved hver udførelse af [f] [DSE] trækkes ***yy*** fra ***nnnnn.*** Derefter undersøger programmet, om ***nnnnn*** er mindre end eller lig med ***xxx***; hvis det er tilfældet, springes næste linie over (den *skippes*), og programmet fortsætter kørslen med den følgende instruktion.



	nnnnn/xxx **i R_I ved start**	**Eksempler på ændring af *nnnnn/xxx* ved gennemløb af løkker**			
Gennemløb →	**0**	**1**	**2**	**3**	**4**
[ISG] →	0.00602	2.00602	4.00602	6.00602	8.00602 (Skip)
[DSE] →	6.00002	4.00002	2.00002	0.00002 (Skip)	-2.00002 (Skip)

Indeksregistrets indirekte funktioner

De indirekte funktioner ændrer ikke indholdet af R_I, derimod bruges heltalsdelen ***(nnnnn)*** som adresse for den funktion, der skal udføres. Indirekte adressering kan med fordel bruges sammen med [f] [ISG] og [f] [DSE] ved adressering af dataregistre, forgreninger og underprogrammer i tilfælde, hvor man skal udføre operationer med adresser i en fast stigende eller faldende rækkefølge.

Indirekte lagring og fremkald. [STO] [(i)] og [RCL] [(i)] * lagrer og fremkalder tal i og fra det dataregister, hvis adresse er den numeriske værdi af ***nnnnn.***
Til adressering af registrene R_0-R_9 bruges ***nnnnn*** = 0-9, til registrene $R_{.0}$-$R_{.9}$ bruges ***nnnnn*** = 10-19, og R_I selv svarer til ***nnnnn*** = 20. (Se også tabellen »Indirekte adressering« side 125.)

Indirekte ombytning med X. [f] [$x \gtrless$ (i)] ombytter indholdet af X-registret med tallet i det dataregister, hvis adresse er den numeriske værdi af ***nnnnn.***

Indirekte registeraritmetik. [STO] ([+], [−], [×], eller [÷]) [(i)] udfører registeraritmetik med det dataregister, hvis adresse er den numeriske værdi af ***nnnnn.***

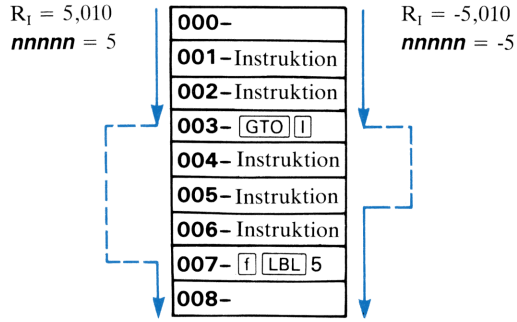
Eksempler gælder:
Hvis 5,01202 lagres i indeksregistret:
nnnnn = 5, ***xxx*** = 012 og ***yy*** = 02
[STO] [(i)] svarer til [STO] 5.
[f] [$x \gtrless$ (i)] svarer til ombytning af indholdet af X og R_5.
[STO] [+] [(i)] svarer til [STO] [+] 5.

* **Se »Forkortet indtastning«, side 74.**

Indirekte hop til etiket eller linie

Under udførelse af et program gælder:

1. Hvis ***nnnnn*** > 0, vil [GTO] [I] få programpilen til at hoppe frem til den næste *etiket* med betegnelsen ***nnnnn.***
2. Hvis ***nnnnn*** < 0, vil [GTO] [I] få programpilen til at hoppe hen til *linien* med nummeret ***nnnnn.***



Indirekte kald af underprogram. Indirekte kald af underprogrammer med [GSB] [I] foregår på samme måde som med [GTO] [I] – blot med den sædvanlige forskel, at der er returhop ved brug af underprogrammer.

Indirekte adressering

nnnnn	[GTO] \ [GSB] [I] giver hop til	[RCL] [(i)] \ [STO] [(i)] påvirker dataregister
0	[f] [LBL] 0	R_0
⋮	⋮	⋮
9	[f] [LBL] 9	R_9
10	" " [A]	$R_{.0}$
11	" " [B]	$R_{.1}$
12	" " [C]	$R_{.2}$
13	" " [D]	$R_{.3}$
14	" " [E]	$R_{.4}$
15	—	$R_{.5}$
16	—	$R_{.6}$
17	—	$R_{.7}$
18	—	$R_{.8}$
19	—	$R_{.9}$
20	—	R_I

Eksempel på brug af løkke med [ISG]

Det følgende program illustrerer brugen af [f] [ISG]. Det indeholder en løkke, der bruger pauser til først at udlæse indholdet af R_I og dernæst kvadratet på dette tal. [f] [ISG] kontrollerer udførelsen af løkken, der skal beregne kvadrattallene for alle de lige tal fra 2 til og med 50.

Indtastning	Lyspanel				
[g] [P/R]	*000-*				Skift til programmerings-status.
[f] CLEAR [PRGM]	*000-*				Slet program-hukommelsen.
[f] [LBL] [A]	*001-*	*42,*	*21,*	*11*	Program »A«.
[f] [FIX] 5	*002-*	*42,*	*7,*	*5*	
2	*003-*			*2*	
[·]	*004-*			*48*	*nnnnn* = 2.
0	*005-*			*0*	
5	*006-*			*5*	
0	*007-*			*0*	*xxx* = 50.
0	*008-*			*0*	
2	*009-*			*2*	*yy* = 2.
[STO] [I]	*010-*		*44*	*25*	Kontroltallet lagres i R_I.
[f] [LBL] 1	*011-*	*42,*	*21,*	*1*	Løkke.
[RCL] [I]	*012-*		*45*	*25*	Tallet i R_I hentes til X.
[g] [INT]	*013-*		*43*	*44*	Heltalsdelen af tallet.
[f] [PSE]	*014-*		*42*	*31*	Heltalsdelen udlæses under en pause.
[g] [x^2]	*015-*		*43*	*11*	Tallet kvadreres.
[f] [PSE]	*016-*		*42*	*31*	Kvadratet udlæses under en pause.
[f] [ISG]	*017-*		*42*	*6*	Træk 2 fra tallet i R_I og test, om tælleren er større end slutværdien (50). Hvis den er større, skippes næste
[GTO] 1	*018*		*22*	*1*	linie, ellers gentages løkken fra etiket 1.
[g] [RTN]	*019*		*43*	*32*	Afslutning på programmet.
[g] [P/R]					Skift til normalstatus.

Indtastning	Lyspanel	
f A	***2,00000***	Når HP-11C starter på
	4,00000	udførelsen af programmet,
		pauseudlæser den først det
	4,00000	lige heltal, der skal
	16,00000	kvadreres, dernæst
	.	udlæses kvadratet i en ny
	.	pause.
	.	Når løkketælleren kommer
	50,00000	over 50, stopper programmet
	2.500,00000	automatisk.

Under programudførelsen sker følgende:

1. Under etiketten »A« lagres tallet 2,05002 som kontroltal i indeksregistret. Formatet er det sædvanlige for løkketællere:

nnnnn	***xxx***	***yy***
(0000)2 Aktuel tællerværdi	050 Testværdi	02 Inkrementværdi

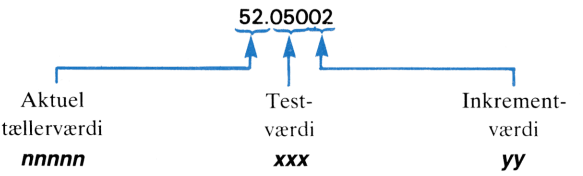
2. Under etiketten »1« udføres følgende:
   Når 2 og 4 (kvadratet på 2) er blevet udlæst, lægges 2 ***(yy)*** til den aktuelle tællerværdi ***(nnnnn),*** som herved bliver 4. Det betyder, at indeksregistret nu indeholder tallet 4,05002, som HP-11C fortolker som:

nnnnn	***xxx***	***yy***
(0000)4 Aktuel tællerværdi	050 Testværdi	02 Inkrementværdi

   Den nye tællerværdi sammenlignes herefter med testværdien 50 ***(xxx).*** Da tælleren ikke er større end testværdien, vil programpilen fortsætte til næste linie, der giver et hop tilbage til løkkens start.

3. Når de første 25 lige heltal (2-50) og deres kvadrater er blevet udlæst, kommer tællerværdien op over 50. Det får programpilen til at overspringe (skippe) linien umiddelbart efter f ISG, så den ender ved linie 019, der indeholder g RTN. I dette tilfælde er g RTN ikke afslutningen på et underprogram, så programpilen hopper op til linie

000, og kontrollen overgives til tastaturet. Når programmet er stoppet, kan De udlæse kontroltallet i indeksregistret med [RCL] [I] :



Tryk [f] [FIX] 4, så De atter får fastkomma 4.

Grænser for brug af [f] [ISG] **og** [f] [DSE]. Bemærk, at [f] [ISG] og [f] [DSE] kan bruges til at inkrementere eller dekrementere ethvert tal, som kan udlæses af HP-11C. Men da lyspanelregistret ikke kan holde tal på mere end 10 cifre, vil decimaldelen af kontroltallet blive påvirket, hvis heltalsdelen ***(nnnnn)*** er på mere end 5 cifre.

Hvis man f.eks. har 99.950,50055 som kontroltal, vil det efter [f] [ISG] være blevet 100.005,50055 – som afrundes til 100.005,5006. HP-11C vil altid betragte de tre første cifre efter kommaet som ***xxx*** og det fjerde ciffer som tierne i ***yy;*** altså bliver det nye kontroltal fortolket, som om ***yy*** er lig 60.

Indirekte registeroperationer

Udførelse af	Eksempel	Lyspanel
3 lagres i R_I, dvs. ***nnnnn*** = 3	3 [STO] [I]	***3,0000***
Lagring og fremkald af $\sqrt{7}$ under brug af R_3 (idet adressen hentes indirekte).	7 [$\sqrt{x}$]	***2,6458***
	[STO] [(i)]	***2,6458***
	[←]	***0,0000***
	[RCL] [(i)]	***2,6458***
	[←]	***0,0000***

Udførelse af	Eksempel	Lyspanel
Indholdet af X-registret ombyttes med tallet i R_3, (idet adressen hentes indirekte).	f x≷(i) f x≷(i)	*2,6458* *0,0000*
Indirekte registeraritmetik med R_3 – hvis indhold divideres med π.	fπ STO ÷ (i)	*3,1416* *3,1416*

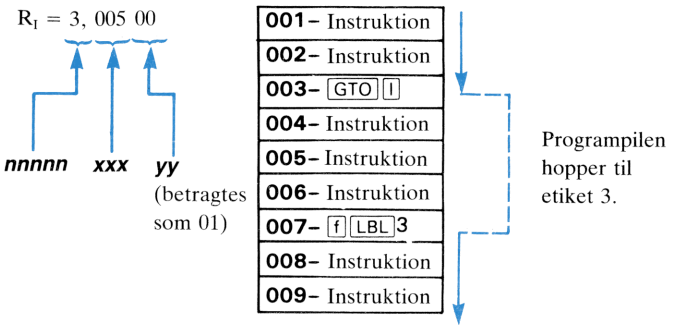
Indirekte registeraritmetik diskuteres nærmere i kapitel 3, afsnit 2 under »f ISG med RCL (i)«.

Indirekte programkontrol

Indirekte adressering af etiketter

Etiketter kan adresseres indirekte på samme måde som dataregistre. Oversigten »Indirekte adressering« side 125 viser, hvilke numeriske adresser der svarer til de mulige etiketter. En sådan adresse er heltalsdelen af et *positivt* kontroltal i indeksregistret.

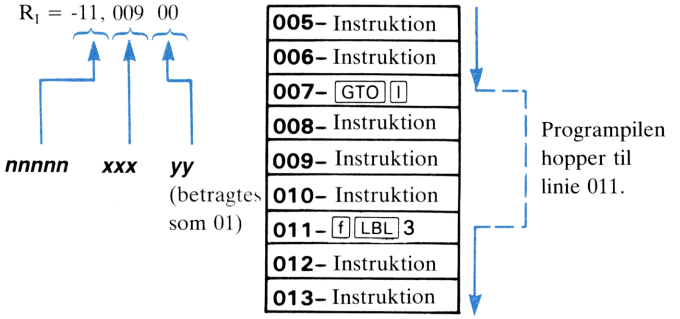
GTO I bruges til indirekte forgrening (hop) til etiketten ***nnnnn.*** Programpilen søger efter ***nnnnn*** på den sædvanlige måde: fra hopordren og frem i programmet. Hvis f.eks. 3,005 er lagret i indeksregistret, får man følgende:



Samme metode bruges til indirekte kald af et underprogram, idet man dog naturligvis skal bruge [GSB] [I] .

Indirekte adressering af linienumre

Linienumre adresseres indirekte med et *negativt* kontroltal i indeksregistret. Som adresse bruges heltalsdelen af kontroltallets numeriske værdi. Forgreningen kan være med eller uden returhop afhængig af, om man bruger [GSB] eller [GTO] for [I] .
Hvis man f.eks. har lagret -11,009 i R_I, fås:



Kapitel 3

HP-11C
Programsamling

Afsnit 1

Programsamling

Matrixoperationer

Programmet beregner determinanten og inverterer en 3×3 matrix, A, (hvis den inverse matrix eksisterer, dvs. hvis Det A $\neq$ 0). Ved hjælp af et mindre kunstgreb kan programmet bruges til at løse det lineære ligningssystem, der har matricen som koefficientmatrix.

Ligninger:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$C = \frac{1}{\det A} \begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & - \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ - \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & - \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & - \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

Hvor C er A inverteret

og $\det A = a_{11}(a_{22}a_{23} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$

og $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$

AX = B løses ved hjælp af X = CB, dvs.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{\det A} \begin{bmatrix} b_1 \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & c_{33} \end{vmatrix} - b_2 \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + b_3 \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -b_1 \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + b_2 \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} - b_3 \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ b_1 \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} - b_2 \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + b_3 \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

Bemærk: Matrixoperationer er særligt udsatte for afrundingsfejl, som i nogle tilfælde kan sløre resultatet i en sådan grad, at det ikke kan bruges.

Hvis De skal løse flere ligningssystemer, kan det bedst betale sig at indtaste determinantrutinen og bestemme determinanten for hver enkelt 3×3 matrix. Derefter erstattes determinantrutinen med ligningsløseren, og den enkelte koefficientmatrices determinant lagres i R_0.

En 2×2 matrix skal lagres som $\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Error 0 vil blive vidst hvis matricen er singulær.

Hovedprogram

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000-	RCL · 1	027- 45 .1
f LBL 9	001-42,21, 9	RCL · 0	028- 45 .0
f ISG	002- 42 6	RCL 8	029- 45 8
f PSE	003- 42 31	RCL · 3	030- 45 .3
RCL (i)	004- 45 24	GSB 8	031- 32 8
×	005- 20	RCL 5	032- 45 5
×	006- 20	RCL · 3	033- 45 .3
+	007- 40	RCL · 1	034- 45 .1
STO 0	008- 44 0	RCL 7	035- 45 7
g RTN	009- 43 32	GSB 8	036- 32 8
f LBL B	010-42,21,12	RCL 8	037- 45 8
RCL 9	011- 45 9	RCL 7	038- 45 7
RCL · 3	012- 45 .3	RCL 5	039- 45 5
RCL · 2	013- 45 .2	RCL · 0	040- 45 .0
RCL · 0	014- 45 .0	GSB 8	041- 32 8
GSB 8	015- 32 8	GSB 2	042- 32 2
RCL · 2	016- 45 .2	RCL 8	043- 45 8
RCL 7	017- 45 7	RCL · 2	044- 45 .2
RCL 6	018- 45 6	RCL · 1	045- 45 .1
RCL · 3	019- 45 .3	RCL 9	046- 45 9
GSB 8	020- 32 8	GSB 8	047- 32 8
RCL 6	021- 45 6	RCL · 1	048- 45 .1
RCL · 0	022- 45 .0	RCL 6	049- 45 6
RCL 9	023- 45 9	RCL 5	050- 45 5
RCL 7	024- 45 7	RCL · 2	051- 45 .2
GSB 8	025- 32 8	GSB 8	052- 32 8
GSB 2	026- 32 2	RCL 5	053- 45 5

TASTER	LYSPANEL	TASTER	LYSPANEL
RCL 9	054- 45 9	R/S	067- 31
RCL 8	055- 45 8	g RTN	068- 43 32
RCL 6	056- 45 6	f LBL 2	069-42,21, 2
f LBL 8	057-42,21, 8	RCL 4	070- 45 4
×	058- 20	g F? 0	071-43, 6, 0
R↓	059- 33	R/S	072- 31
×	060- 20	f LBL 3	073-42,21, 3
g R↑	061- 43 33	1	074- 1
−	062- 30	STO I	075- 44 25
RCL 0	063- 45 0	0	076- 0
÷	064- 10	STO 4	077- 44 4
g F? 0	065-43, 6, 0	g RTN	078- 43 32
GTO 1	066- 22 1		

REGISTRE			R_i: Indeks
R_0: Det A	R_1: b_1	R_2: b_2	R_3: b_3
R_4: x_1, x_2, x_3	R_5: a_{11}	R_6: a_{12}	R_7: a_{13}
R_8: a_{21}	R_9: a_{22}	$R_{.0}$: a_{23}	$R_{.1}$: a_{31}
$R_{.2}$: a_{32}	$R_{.3}$: a_{33}		

Determinantrutine

TASTER	LYSPANEL	TASTER	LYSPANEL
f LBL A	079-42,21,11	RCL · 0	083- 45 .0
* 4	080- 4	RCL · 2	084- 45 .2
STO I	081- 44 25	GSB 9	085- 32 9
0	082- 0	RCL 8	086- 45 8

* Indsæt g CF 0 efter linie 079

TASTER	LYSPANEL	TASTER	LYSPANEL
RCL · 3	087- 45 .3	GSB 9	095- 32 9
GSB 9	088- 32 9	RCL 5	096- 45 5
RCL 9	089- 45 9	RCL · 3	097- 45 .3
RCL · 1	090- 45 .1	GSB 9	098- 32 9
GSB 9	091- 32 9	RCL 6	099- 45 6
CHS	092- 16	RCL · 1	100- 45 .1
RCL 7	093- 45 7	GSB 9	101- 32 9
RCL · 2	094- 45 .2		

Ligningsrutine

f LBL 1	079-42,21, 1	STO 3	088- 44 3
RCL (i)	080- 45 24	R↓	089- 33
×	081- 20	STO 2	090- 44 2
STO + 4	082-44,40, 4	R↓	091- 33
f ISG	083- 42 6	STO 1	092- 44 1
f PSE	084- 42 31	GSB 3	093- 32 3
g RTN	085- 43 32	GSB B	094- 32 12
f LBL C	086-42,21,13	g CF 0	095-43, 5, 0
g SF 0	087-43, 4, 0	RCL 4	096- 45 4

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast hovedprogrammet			
	(linierne 00-078).			
2	Determinant			
	Indtast rutinen f LBL A			
	(linierne 079-101).		GTO · 078	
			g P/R ... g P/R	

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
3	Vælg USER-status.			
4	Matricens elementer lagres			
	a_{11}	a_{11}	STO 5	
	a_{12}	a_{12}	STO 6	
	a_{13}	a_{13}	STO 7	
	a_{21}	a_{21}	STO 8	
	a_{22}	a_{22}	STO 9	
	a_{23}	a_{23}	STO · 0	
	a_{31}	a_{31}	STO · 1	
	a_{32}	a_{32}	STO · 2	
	a_{33}	a_{33}	STO · 3	
5	Find determinanten.		A	Det A
6	A^{-1}			
	Find den inverse matrix.		B	C_{11}
			R/S	C_{12}
			R/S	C_{13}
			R/S	C_{21}
			R/S	C_{22}
			R/S	C_{23}
			R/S	C_{31}
			R/S	C_{32}
			R/S	C_{33}
7	Lineært ligningssystem			
	Slet linierne 079-101.		g RTN g BST	
			g P/R	101- 32 9
			←... ←	078- 43 32
8	Indtast ligningsrutinen f		f LBL 1	079-42,21, 1
	(linierne 079-096).		... g P/R	

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
9	Indlæs søjlevektoren	b_1	ENTER	
	og find x.	b_2	ENTER	
		b_3	C	x_1
			R/S	x_2
			R/S	x_3
10	Gentag TRIN 9 for ny			
	søjlevektor.			
11	Slet linierne 079-096 for			
	ny determinant		g RTN g BST	
			g P/R	096- 45 4
			←...←	078- 43 32
	og fortsæt fra TRIN 2.			

Eksempel 1: Invertér $\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix}$ og løs dernæst

$$\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 5 \end{bmatrix}$$

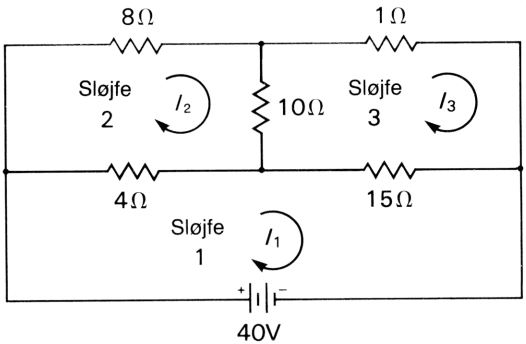
Indtastning	**Lyspanel**	
Vælg USER-status		
f FIX 4		
Indtast determinantrutinen		
14 STO 5		
8 CHS STO 6		
0 STO 7		
8 CHS STO 8		
12 STO 9		
0 STO · 0		
STO · 1 STO · 2		
1 STO · 3		
A	**104.0000**	Det. A
B	**0.1154**	C_{11}

Indtastning	Lyspanel	
R/S	0.0769	C_{12}
R/S	0.0000	C_{13}
R/S	0.0769	C_{21}
R/S	0.1346	C_{22}
R/S	0.0000	C_{23}
R/S	0.0000	C_{31}
R/S	0.0000	C_{32}
R/S	1.0000	C_{33}
g RTN g BST		
g P/R	101- 32 9	
← ← ... ←	078- 43 32	

Indtast ligningsrutinen.

Indtastning	Lyspanel	
g P/R		
20 ENTER 5 ENTER		
0 C	2.6923	x_1
R/S	2.2115	x_2
R/S	0.0000	x_3

Eksempel 2:
Find cirkulationsstrømmene i følgende kredsløb.



De tre strømligninger er:

Sløjfe 1 $$4I_1 - 4I_2 + 15I_1 - 15I_3 - 40 = 0$$

Sløjfe 2 $$4I_2 - 4I_1 + 8I_2 + 10I_2 - 10I_3 = 0$$

Sløjfe 3 $$10I_3 - 10I_2 + 1I_3 + 15I_3 - 15I_1 = 0$$

eller

$$19I_1 - 4I_2 - 15I_3 = 40$$
$$-4I_1 + 22I_2 - 10I_3 = 0$$
$$-15I_1 - 10I_2 + 26I_3 = 0$$

på matrixform

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}.$$

Indtastning	**Lyspanel**
19 [STO] 5	
4 [CHS] [STO] 6	
15 [CHS] [STO] 7	
4 [CHS] [STO] 8	
22 [STO] 9	
10 [CHS] [STO] [·] 0	
15 [CHS] [STO] [·] 1	
10 [CHS] [STO] [·] 2	
26 [STO] [·] 3	

Slet ligningsrutinen, hvis den stadig er lagret i programhukommelsen og indtast determinant rutinen.

[A]	**2,402.0000**

Slet determinantrutinen og indtast ligningsrutinen.

40 [ENTER]	
0 [ENTER] [C]	**7.8601**
[R/S]	**4.2298**
[R/S]	**6.1615**

3 ligninger med 3 ubekendte

Følgende program løser 3 ligninger med 3 ubekendte under brug af Cramer's formel.
Programmet er taget med, fordi det er lettere at bruge ved ren løsning af lineære ligningssystemer end det generelle matrixprogram.

Formler:

$$AX = B.$$

3 ubekendte:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Systemets determinant:

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) .$$

x_i'erne findes af $x_i = \dfrac{\det(i)}{\text{Det A}}$ for Det A $\neq 0$,

hvor det(i) er determinanten for A med den i'ende søjle erstattet af b.

Bemærkninger:
Hvis Det A = 0, er systemet ikke lineært uafhængigt, og programmet kan ikke bruges, og kørslen afsluttes af **Error 0.** Hvis Det A er meget nær 0, er problemet dårligt formuleret, og HP-11C's repræsentation af Det A vil være usikker. Tilsvarende bør man lade troen på forholdet det(i)/Det A falde i takt med antallet af betydende cifre i Det A.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000-	RCL 5	029- 45 5
f LBL A	001-42,21,11	RCL 7	030- 45 7
1	002- 1	GSB 9	031- 32 9
.	003- 48	CHS	032- 16
0	004- 0	RCL 3	033- 45 3
0	005- 0	RCL 8	034- 45 8
9	006- 9	GSB 9	035- 32 9
STO I	007- 44 25	RCL 1	036- 45 1
f LBL 2	008-42,21, 2	RCL 9	037- 45 9
RCL I	009- 45 25	GSB 9	038- 32 9
g INT	010- 43 44	RCL 2	039- 45 2
f PSE	011- 42 31	RCL 7	040- 45 7
RCL (i)	012- 45 24	f LBL 9	041-42,21, 9
R/S	013- 31	f DSE	042- 42 5
STO (i)	014- 44 24	f PSE	043- 42 31
f ISG	015- 42 6	RCL (i)	044- 45 24
GTO 2	016- 22 2	×	045- 20
GSB 0	017- 32 0	×	046- 20
STO 0	018- 44 0	+	047- 40
R/S	019- 31	g RTN	048- 43 32
f LBL 0	020-42,21, 0	f LBL B	049-42,21,12
0	021- 0	STO . 3	050- 44 .3
STO I	022- 44 25	R↓	051- 33
RCL 6	023- 45 6	STO . 2	052- 44 .2
RCL 8	024- 45 8	R↓	053- 33
GSB 9	025- 32 9	STO . 1	054- 44 .1
RCL 4	026- 45 4	1	055- 1
RCL 9	027- 45 9	GSB 7	056- 32 7
GSB 9	028- 32 9	R/S	057- 31

TASTER	LYSPANEL	TASTER	LYSPANEL
4	058– 4	RCL · 0	075– 45 .0
GSB 7	059– 32 7	g RTN	076– 43 32
R/S	060– 31	f LBL 8	077–42,21, 8
7	061– 7	RCL · 1	078– 45 .1
GSB 7	062– 32 7	f $x \geqslant (i)$	079– 42 23
R/S	063– 31	STO · 1	080– 44 .1
f LBL 7	064–42,21, 7	f ISG	081– 42 6
STO · 4	065– 44 .4	f PSE	082– 42 31
STO I	066– 44 25	RCL · 2	083– 45 .2
GSB 8	067– 32 8	f $x \geqslant (i)$	084– 42 23
GSB 0	068– 32 0	STO · 2	085– 44 .2
RCL 0	069– 45 0	f ISG	086– 42 6
÷	070– 10	f PSE	087– 42 31
STO · 0	071– 44 .0	RCL · 3	088– 45 .3
RCL · 4	072– 45 .4	f $x \geqslant (i)$	089– 42 23
STO I	073– 44 25	STO · 3	090– 44 .3
GSB 8	074– 32 8	g RTN	091– 43 32

REGISTRE			R_I: Indeks
R_0: Det	R_1: a_{11}	R_2: a_{21}	R_3: a_{31}
R_4: a_{12}	R_5: a_{22}	R_6: a_{32}	R_7: a_{13}
R_8: a_{23}	R_9: a_{33}	$R_{.0}$: det (i)	$R_{.1}$: b_1
$R_{.2}$: b_2	$R_{.3}$: b_3	$R_{.4}$: Index	

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Vælg USER-status.			

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
3	Indtast 3×3 matricen. (Bemærk:		[A]	1, R_1
	en allerede lagret værdi	a_{11}	[R/S]	2, R_2
	genlagres ved blot at trykke	a_{21}	[R/S]	3, R_3
	på [R/S].)	a_{31}	[R/S]	4, R_4
		a_{12}	[R/S]	5, R_5
		a_{22}	[R/S]	6, R_6
		a_{32}	[R/S]	7, R_7
		a_{13}	[R/S]	8, R_8
		a_{23}	[R/S]	9, R_9
		a_{33}	[R/S]	Det.
	Matricen genudlæses ved tryk på [A]			
4	Indtast søjlevektoren.	b_1	[ENTER]	b_1
		b_2	[ENTER]	b_2
		b_3	[B]	x_1
			[R/S]	x_2
			[R/S]	x_3
5	Gentag TRIN 4 ved indtastning af ny søjlevektor.			
6	Gentag fra TRIN 3 ved beregning af nyt 3×3-system.			

Eksempel 1:
Find løsningen til følgende ligningssystem:

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}$$

Indtastning	**Lyspanel**	
[f] CLEAR [REG]		
Vælg USER-status		
[f] [FIX] 4		
[A]	**1.0000**	**0.0000**

Indtastning	Lyspanel		
19 R/S	2.0000	0.0000	
4 CHS R/S	3.0000	0.0000	
15 CHS R/S	4.0000	0.0000	
4 CHS R/S	5.0000	0.0000	
22 R/S	6.0000	0.0000	
10 CHS R/S	7.0000	0.0000	
15 CHS R/S	8.0000	0.0000	
10 CHS R/S	9.0000	0.0000	
26 R/S	2,402.0000		Det.
40 ENTER			
0 ENTER B	7.8601		x_1
R/S	4.2298		x_2
R/S	6.1615		x_3

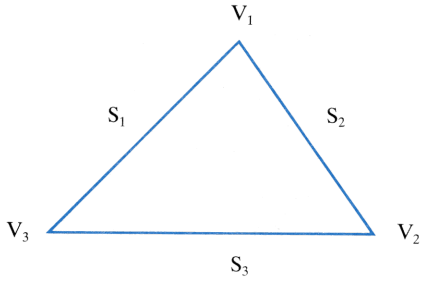
Eksempel 2:
Find løsningen til:

$$\begin{bmatrix} 19 & -4 & 4 \\ 5 & -12 & -10 \\ -15 & 8 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 4 \end{bmatrix}$$

Indtastning	Lyspanel		
A	1.0000	19.0000	
R/S	2.0000	-4.0000	
5 R/S	3.0000	-15.0000	
R/S	4.0000	-4.0000	
R/S	5.0000	22.0000	
12 CHS R/S	6.0000	-10.0000	
8 R/S	7.0000	-15.0000	
4 R/S	8.0000	-10.0000	
R/S	9.0000	26.0000	
3 R/S	-264.0000		Det.
5 ENTER			
3 CHS ENTER			
4 B	-1.6667		x_1
R/S	-4.4091		x_2
R/S	4.7576		x_3

Trekantsberegninger

Dette program kan bruges til at beregne sider, vinkler og areal for enhver plan trekant.



Det vil i de fleste tilfælde være nok at specificere tre af de seks trekantsparametre (tre sider og tre vinkler), om end der gives tilfælde, hvor tre vinkler ikke specificerer trekanten entydigt.
Programmet kan således bruges til de fem sædvanlige trekantstilfælde:

1. Tre sider (SSS).
2. En side og de to hosliggende vinkler **(VSV)**.
3. En side, dens modstående og en af de hosliggende vinkler **(SVV)**.
4. En vinkel og dens to hosliggende sider **(SVS)**.
5. En vinkel, dens modstående og en af de hosliggende sider **((SSV)** - kan være flertydig).

Hvis de tre værdier, der bruges som inddata, vælges i retningen med uret, får man tilsvarende udlæst resultaterne med uret. Resultaterne lagres i dataregistrene R_0 til R_6:

Areal	Register 0
Side 1	Register 1
Vinkel 1	Register 2
Side 2	Register 3
Vinkel 2	Register 4
Side 3	Register 5
Vinkel 3	Register 6

Bemærkninger:

- Inddata kan være i et vilkårligt vinkelmål (grader, nygrader eller radian), og brugeren må selv sørge for, at vinkelmålet passer med de data, der skal behandles.

- Det skal bemærkes, at vinkler og sider af praktiske grunde ikke er benævnt, som man normalt ser det – f.eks. er S_1 *ikke* V_1's modstående side.

- Vinkler skal indlæses i decimalnotation. Vinkler, der er givet i grader/min/sek., kan omregnes til decimalgrader med [g] [→H].

- Der kan optræde betydelige afrundingsfejl ved beregninger på trekanter, der er ekstremt spidsvinklede (hvor mindst én vinkel nærmer sig 0).

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000-	RCL 2	029- 45 2
f LBL A	001-42,21,11	RCL 6	030- 45 6
STO 5	002- 44 5	+	031-- 40
R↓	003- 33	SIN	032- 23
STO 3	004- 44 3	÷	033- 10
R↓	005- 33	RCL 1	034- 45 1
STO 1	006- 44 1	×	035- 20
RCL 3	007- 45 3	STO 3	036- 44 3
g →P	008- 43 26	GTO 0	037- 22 0
g x^2	009- 43 11	f LBL C	038-42,21,13
RCL 5	010- 45 5	STO 4	039- 44 4
g x^2	011- 43 11	R↓	040- 33
−	012- 30	STO 2	041- 44 2
RCL 1	013- 45 1	R↓	042- 33
RCL 3	014- 45 3	STO 1	043- 44 1
×	015- 20	RCL 4	044- 45 4
2	016- 2	RCL 2	045- 45 2
×	017- 20	+	046- 40
÷	018- 10	SIN	047- 23
g COS^{-1}	019- 43 24	RCL 4	048- 45 4
STO 2	020- 44 2	SIN	049- 23
GTO 0	021- 22 0	÷	050- 10
f LBL B	022-42,21,12	RCL 1	051- 45 1
STO 2	023- 44 2	×	052- 20
R↓	024- 33	STO 3	053- 44 3
STO 1	025- 44 1	GTO 0	054- 22 0
R↓	026- 33	f LBL E	055-42,21,15
STO 6	027- 44 6	STO 4	056- 44 4
SIN	028- 23	R↓	057- 33

TASTER	LYSPANEL	TASTER	LYSPANEL
STO 3	058– 44 3	f LBL D	087–42,21,14
R↓	059– 33	STO 3	088– 44 3
STO 1	060– 44 1	R↓	089– 33
RCL 3	061– 45 3	STO 2	090– 44 2
RCL 4	062– 45 4	R↓	091– 33
SIN	063– 23	STO 1	092– 44 1
RCL 1	064– 45 1	f LBL 0	093–42,21, 0
÷	065– 10	RCL 2	094– 45 2
×	066– 20	RCL 1	095– 45 1
g SIN⁻¹	067– 43 23	f →R	096– 42 26
RCL 4	068– 45 4	RCL 3	097– 45 3
+	069– 40	$x \gtrless y$	098– 34
GSB 9	070– 32 9	−	099– 30
STO 2	071– 44 2	g →P	100– 43 26
GSB 0	072– 32 0	STO 5	101– 44 5
RCL 1	073– 45 1	$x \gtrless y$	102– 34
RCL 3	074– 45 3	STO 4	103– 44 4
f $x \leqslant y$	075– 42 10	RCL 2	104– 45 2
GTO 8	076– 22 8	+	105– 40
2	077– 2	GSB 9	106– 32 9
R/S	078– 31	STO 6	107– 44 6
RCL 6	079– 45 6	SIN	108– 23
GSB 9	080– 32 9	×	109– 20
STO 6	081– 44 6	RCL 1	110– 45 1
RCL 4	082– 45 4	×	111– 20
+	083– 40	2	112– 2
GSB 9	084– 32 9	÷	113– 10
STO 2	085– 44 2	STO 0	114– 44 0
GTO 0	086– 22 0	GTO 1	115– 22 1

TASTER	LYSPANEL	TASTER	LYSPANEL
[g] [RTN]	116- 43 32	6	126- 6
[f] [LBL] 9	117-42,21, 9	[STO] [I]	127- 44 25
[COS]	118- 24	[f] [LBL] 2	128-42,21, 2
[CHS]	119- 16	[RCL] [(i)]	129- 45 24
[g] [COS^{-1}]	120- 43 24	[R/S]	130- 31
[g] [RTN]	121- 43 32	[f] [ISG]	131- 42 6
[f] [LBL] 1	122-42,21, 1	[GTO] 2	132- 22 2
[.]	123- 48	[f] [LBL] 8	133-42,21, 8
0	124- 0	[g] [CLx]	134- 43 35
0	125- 0		

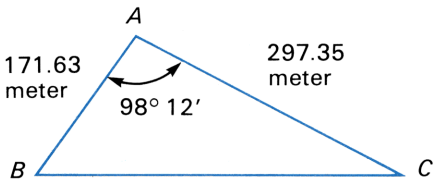
REGISTRE			R_i: Indeks
R_0: Areal	R_1: S_1	R_2: A_1	R_3: S_2
R_4: A_2	R_5: S_3	R_6: A_3	R_7—R_8: Ubrugt

1	Indtast programmet.			
2	Vælg USER-status.			
3	Vælg vinkelmål.			
	[DEG, RAD, GRAD].			

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
4	SSS (3 sider)			
	Indtast:			
	Side 1	S_1	ENTER	
	Side 2	S_2	ENTER	
	Side 3	S_3	A	AREAL
	Fortsæt med TRIN 9.			
5	VSV (side + to hosliggende vinkler)			
	Indtast:			
	Vinkel 3	V_3	ENTER	
	Side 1	S_1	ENTER	
	Vinkel 1	V_1	B	AREAL
	Fortsæt med TRIN 9.			
6	SVV (side, modstående og			
	hosliggende vinkel)			
	Side 1	S_1	ENTER	
	Vinkel 1	V_1	ENTER	
	Vinkel 2	V_2	C	AREAL
	Fortsæt med TRIN 9.			
7	SVS (vinkel og to hosliggende sider)			
	Indtast:			
	Side 1	S_1	ENTER	
	Vinkel 1	V_1	ENTER	
	Side 2	S_2	D	AREAL
	Fortsæt med TRIN 9.			
8	SSV (vinkel, modstående			
	og hosliggende side) Indtast:			
	Side 1	S_1	ENTER	
	Side 2	S_2	ENTER	

	Vinkel 2	V_2	E	AREAL
9	Sider og vinkler:			
	Udlæsning af			
	sider og vinkler:		R/S	S_1
			R/S	V_1
			R/S	S_2
			R/S	V_2
			R/S	S_3
			R/S	V_3
9a	Hvis løsningen er entydig:		R/S	0
9b	Hvis der er to løsninger:		R/S	2
			R/S	AREAL
			R/S	S_1
			R/S	V_1
			R/S	S_2
			R/S	V_2
			R/S	S_3
			R/S	V_3

Eksempel 1:
En landmåler skal opmåle en trekantet parcel. Længderne AB og AC måles med et elektronisk instrument. Tilsvarende måles vinklen BAC.



Dette er en Side-Vinkel-Side-opgave, og tallene opgives at være:

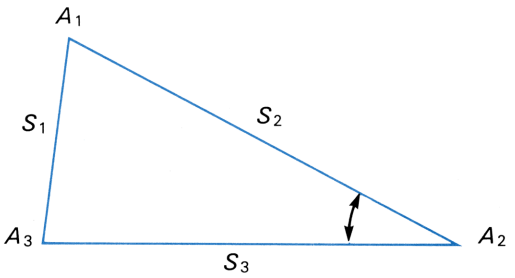
$S_1 = 171{,}63$; $V_1 = 98°12'$; $S_2 = 297{,}35$

Indtastning	**Lyspanel**	
[g] [DEG]		
[f] [FIX] 2		
Vælg USER-status		
171.63 [ENTER]	**171.63**	
98.12 [g] [→H]	**98.20**	
297.35 [D]	**25,256.21**	Areal
[R/S]	**171.63**	S_1
[R/S]	**98.20**	V_1
[R/S]	**297.35**	S_2
[R/S]	**27.83**	V_2
[R/S]	**363.91**	S_3
[R/S]	**53.97**	V_3

Eksempel 2:

Der er givet en vinkel, dens modstående og en af de hosliggende sider. Bestem de resterende sider og vinkler samt arealet.

$S_1 = 25{,}6$
$S_2 = 32{,}8$
$V_2 = 42{,}3°$



Indtastning	Lyspanel	
25.6 [ENTER]	**25.60**	
32.8 [ENTER]	**32.80**	
42.3 [E]	**410.85**	Areal
[R/S]	**25.60**	S_1
[R/S]	**78.12**	V_1
[R/S]	**32.80**	S_2
[R/S]	**42.30**	V_2
[R/S]	**37.22**	S_3
[R/S]	**59.58**	V_3
[R/S]	**2.00**	Anden løsning
[R/S]	**124.68**	Areal
[R/S]	**25.60**	S_1
[R/S]	**17.28**	V_1
[R/S]	**32.80**	S_2
[R/S]	**42.30**	V_2
[R/S]	**11.30**	S_3
[R/S]	**120.42**	V_3

Kurvetilpasning

HP-11C er forsynet med lineær regression som fastfunktion. Funktionen kan direkte bruges til lineær regression, og den kan, ved hjælp af simple variabeltransformationer, også bruges til andre former for regressionsanalyse.

Programmet kan bruges til regressionsanalyse med følgende kurvetyper, som alle kan transformeres til den generelle lineære form: $Y = A + BX$

1. Eksponentialkurver: $Y = ae^{bx}$ $(a>0)$
2. Logaritmekurver: $y = a + b \ln x$
3. Potenskurver: $y = ax^b$ $(a>0)$

Regressionskoefficienterne A og b findes ved at løse følgende lineære ligningssystem:

$$\begin{bmatrix} n & \Sigma X_i \\ \Sigma X_i & \Sigma X_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \Sigma Y_i \\ \Sigma (Y_i X_i) \end{bmatrix}$$

hvor n er antallet af datapar

Variabeltransformationerne er foretaget som vist i nedenstående tabel.

Regressionstype	A	X_i	Y_i	Kode
Eksponentiel	ln a	x_i	ln y_i	1
Logaritmisk	a	ln x_i	y_i	2
Potens	ln a	ln x_i	ln y_i	3

Korrelationskoefficientens kvadrat, r^2, er givet ved:

$$r^2 = \frac{A\Sigma Y_i + b\Sigma X_i Y_i - \frac{1}{n}(\Sigma Y_i)^2}{\Sigma(Y_i^2) - \frac{1}{n}(\Sigma Y_i)^2}$$

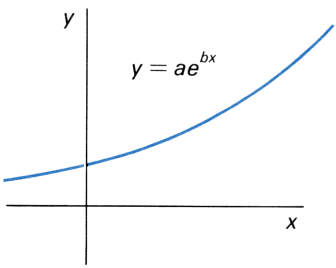
Inden data indlæses, vælges regressionstypen ved hjælp af kodetallet (1-3).

Korrelationskoefficientens kvadrat, r^2, er et mål for kurvens tilpasning til datasættet. Jo nærmere r^2 kommer til 1, desto bedre passer kurven til datasættet. Modsætningsvist vil en værdi nær 0 angive, at den undersøgte sammenhæng ikke kan findes.

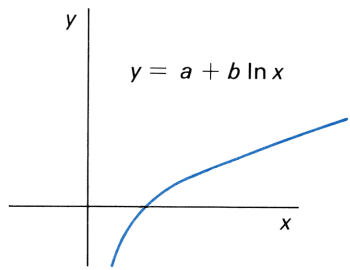
r^2 minder således ganske meget om korrelationskoefficienten, r, som blev omtalt i forbindelse med statistikfunktionerne. En diskussion af forskellene falder uden for denne håndbogs rammer.

Regressionskoefficienterne a og b definerer kurven efter de formler, der blev nævnt i indledningen til dette programeksempel.

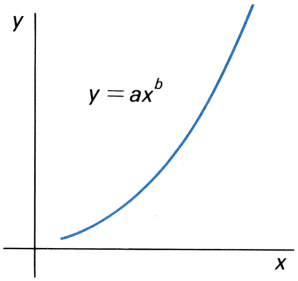
Eksponentialkurve
Kode = 1



Logaritmekurve
Kode = 2



Potenskurve
Kode = 3



Bemærkninger:

- Programmet anvender mindste kvadraters metode til bestemmelse af regressionskoefficienterne. Ved tilpasning med logaritmekurver bruges det originale udtryk (som jo allerede er lineært), hvorimod det transformerede lineære bruges ved tilpasning med eksponential- og potenskurver.
- Bruges data med en x_i-værdi mindre end eller lig 0, gives en sædvanlig fejlmeddelelse ved logaritmisk tilpasning. Bruges en eksponentialkurve, gives tilsvarende fejlmeddelelser for y_i mindre end eller lig 0. I forbindelse med potenskurver skal såvel x_i som y_i være større end 0.
- Ved meget små differenser mellem x_i og/eller y_i-værdier (eller de transformerede) falder nøjagtigheden ved bestemmelsen af regressionskoefficienterne.
- Programmet nulstiller alle dataregistre, så inden man bruger programmet, bør man altid sikre sig, at der ikke er lagret vigtige data i noget dataregister.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000-	6	006- 6
f LBL A	001-42,21,11	+	007- 40
f CLEAR REG	002- 42 34	STO I	008- 44 25
f FIX 2	003-42, 7, 2	R↓	009- 33
ENTER	004- 36	g SF 0	010-43, 4, 0
ENTER	005- 36	g SF 1	011-43, 4, 1

TASTER	LYSPANEL	TASTER	LYSPANEL
GTO I	012- 22 25	g Σ-	036- 43 49
f LBL 7	013-42,21, 7	GTO 9	037- 22 9
g CF 1	014-43, 5, 1	f LBL B	038-42,21,12
GTO 9	015- 22 9	f L.R.	039- 42 49
f LBL 8	016-42,21, 8	g F? 0	040-43, 6, 0
g CF 0	017-43, 5, 0	e^x	041- 12
f LBL 9	018-42,21, 9	R/S	042- 31
R/S	019- 31	$x \gtrless y$	043- 34
g F? 0	020-43, 6, 0	R/S	044- 31
g LN	021- 43 12	f $\hat{y},r$	045- 42 48
$x \gtrless y$	022- 34	$x \gtrless y$	046- 34
g F? 1	023-43, 6, 1	g x^2	047- 43 11
g LN	024- 43 12	g RTN	048- 43 32
RCL 6	025- 45 6	f LBL D	049-42,21,14
g $x \neq 0$	026- 43 30	1	050- 1
GTO 6	027- 22 6	STO 6	051- 44 6
R↓	028- 33	GTO 9	052- 22 9
Σ+	029- 49	f LBL C	053-42,21,13
GTO 9	030- 22 9	g F? 1	054-43, 6, 1
f LBL 6	031-42,21, 6	g LN	055- 43 12
R↓	032- 33	f $\hat{y},r$	056- 42 48
0	033- 0	g F? 0	057-43, 6, 0
STO 6	034- 44 6	e^x	058- 12
R↓	035- 33		

REGISTER			R_i: Kode
R_0: i	R_1: Σx_i	R_2: Σx_i^2	R_3: Σy_i
R_4: Σy_i^2	R_5: $\Sigma x_i y_i$	R_6: 1 eller 0	R_7—R_9: Ubrugt

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Vælg USER-status.			
3	Vælg regressionstype:			
	Eksponentialkurve	1	A	1.00
	Logaritmekurve	2	A	2.00
	Potenskurve	3	A	3.00
4	Indlæs x_i, y_i.	x_i	ENTER	
	(Gentag TRIN 4 for	y_i	R/S	i
	alle (x_i, y_i)-par.)			
5	Beregning af regressions-		B	a
	koefficienter og korrelations-		R/S	b
	koefficientens kvadrat (r^2).		R/S	r^2
6	Beregn $\hat{y}$ (y-estimat) efter			
	indtastning af x-værdi.	x	C	$\hat{y}$
	(Gentag TRIN 6 så tit			
	det ønskes).			
7	Korrektion af fejl. Fejlindtastninger			
	i TRIN 3 kan rettes ved at trykke		D	
	D og genindtaste det forkerte talpar	$x_i$err	ENTER	
	Dernæst fortsættes med TRIN 4,	$y_i$err	R/S	$i-1$
	hvor det rigtige talpar indtastes.			

Eksempel 1: (Eksponentialkurve. Kode = 1)

x_i	0.72	1.31	1.95	2.58	3.14
y_i	2.16	1.61	1.16	0.85	0.5

Løsning: $a = 3.45, \quad b = -0.58$

$$y = 3.45\, e^{-0.58x}$$

$$r^2 = 0.98$$

Indtastning	Lyspanel	
Vælg USER-status		
1 [A]	**1.00**	
.72 [ENTER] 2.16 [R/S]		
1.31 [ENTER] 1.61 [R/S]		
1.95 [ENTER] 1.16 [R/S]		
2.58 [ENTER] .85 [R/S]		
3.14 [ENTER] .5 [R/S]		
[B]	**3.45**	a
[R/S]	**-0.58**	b
[R/S]	**0.98**	r^2
1.5 [C]	**1.44**	$\hat{y}$

Eksempel 2: (Logaritmekurve, Kode = 2)

x_i	3	4	6	10	12
y_i	1.5	9.3	23.4	45.8	60.1

Løsning: $a = -47.02, \quad b = 41.39$

$y = 47.02 + 41.39 \ln x$

$r^2 = 0.98$

For $x = 8, \quad \hat{y} = 39.06$

For $x = 14.5, \quad \hat{y} = 63.67$

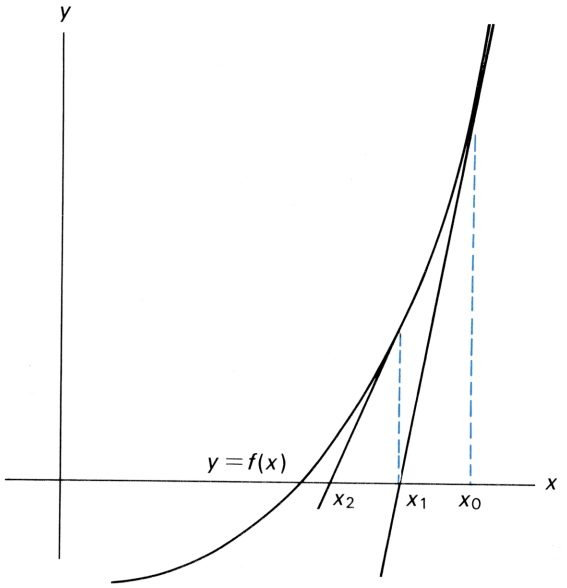
Indtastning	Lyspanel	
2 [A]	**2.00**	
3 [ENTER] 1.5 [R/S]		
4 [ENTER] 9.3 [R/S]		
6 [ENTER] 23.4 [R/S]		
10 [ENTER] 45.8 [R/S]		
12 [ENTER] 60.1 [R/S]		
[B]	**-47.02**	a
[R/S]	**41.39**	b
[R/S]	**0.98**	r^2
8 [C]	**39.06**	$\hat{y}$
14.5 [C]	**63.67**	$\hat{y}$

Newton-Raphson-iteration

Et af de mest almindelige og mest frustrerende problemer indenfor den anvendte matematik er kravet om løsning af ligninger som f.eks.

$$\ln x + 3x = 10{,}8074$$

hvor det er totalt umuligt at få isoleret x på den ene side af lighedstegnet. det betyder, at der ikke er nogen simpel analytisk løsning. Det følgende program anvender Newton-Raphson-iteration til at løse ligninger på formen f (x)=0, hvor den ubekendte, x, er implicit givet.
Brugeren skal selv indtaste en række instruktioner til beregning af funktionen, f (x). Dette underprogram skal lade x være tallet i X-registret og skal desuden respektere, at hovedprogrammet lægger beslag på dataregistrene 0-4 og optager 68 programlinier. Resten af hukommelsen er til fri disposition for brugeren.
Programmet startes ved, at brugeren initialiserer med et gæt på løsningen. Jo bedre gættet er, desto hurtigere konvergerer iterationen.



Programmet fortsætter iterationen, indtil en af følgende betingelser er opfyldt:

1) Løkketæller > Løkkemaksimum
2) $|f(x)| <$ tolerancen, ε
3) $|x_{i+1} - x_i| < \triangle x$

Brugeren kan selv vælge størrelsen af ε og $\triangle$ eller bruge programmets normalværdier.

Formel:
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Programmet tilnærmer funktionens første afledte med følgende udtryk:

$$x_{i+1} = x_i - \delta_i \frac{f(x_i)}{f(x_i + \delta_i) - f(x_i)}$$

hvor $\delta_0 = 10^{-5} x_0$ og $\delta_i = x_i - x_{i-1}$

Bemærk:

- Når programmet stopper, udlæses det beregnede x, som samtidig lagres i R_2.
- Der udlæses en fejlmeddelelse ***(Error 0),*** hvis der divideres med 0 under beregningerne (det vil typisk kunne ske, hvis $f'(x_i)$ nærmer sig 0). Hvis dette sker, kan man prøve et starttal, der ligger lidt ved siden af den sidste tilnærmelse.
- Programmet tæller antallet af gennemløb af løkken (antallet af iterationer). R_I initialiseres til 50 i linierne 020-021. Hvis der kræves mere end 50 iterationer, udlæses en fejlmeddelelse: ***Error 4.*** Brugeren kan selv ændre på maksimum – men hvis der ikke findes en løsning efter 50 iterationer, er der ikke meget håb om at få afsluttet beregningen på det givne grundlag.
- Der kan indsættes en [f] [PSE] -instruktion i linie 047, hvorved den seneste tilnærmelse bliver udlæst.
- Man bør ikke gøre $\triangle x$ mindre end 10^{-8}; mindre værdier vil nemt blive overdøvet af afgrundsfejl, så det bliver svært at fortolke resultatet.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000-	GSB C	029- 32 13
f LBL A	001-42,21,11	STO 2	030- 44 2
EEX	002- 26	RCL 0	031- 45 0
CHS	003- 16	RCL 1	032- 45 1
5	004- 5	+	033- 40
STO 0	005- 44 0	GSB C	034- 32 13
EEX	006- 26	RCL 2	035- 45 2
CHS	007- 16	−	036- 30
8	008- 8	g $x=0$	037- 43 40
STO 3	009- 44 3	STO · 9	038- 44 .9
STO 4	010- 44 4	RCL 2	039- 45 2
R/S	011- 31	$x \gtrless y$	040- 34
STO 3	012- 44 3	÷	041- 10
R/S	013- 31	RCL 0	042- 45 0
STO 4	014- 44 4	×	043- 20
g CLx	015- 43 35	CHS	044- 16
g RTN	016- 43 32	RCL 1	045- 45 1
f LBL B	017-42,21,12	+	046- 40
STO 2	018- 44 2	STO 2	047- 44 2
STO × 0	019-44,20, 0	g LSTx	048- 43 36
5	020- 5	−	049- 30
0	021- 0	STO 0	050- 44 0
STO I	022- 44 25	f DSE	051- 42 5
GSB 1	023- 32 1	GTO 2	052- 22 2
RCL 2	024- 45 2	GSB 9	053- 32 9
g RTN	025- 43 32	f LBL 2	054-42,21, 2
f LBL 1	026-42,21, 1	RCL 2	055- 45 2
RCL 2	027- 45 2	GSB C	056- 32 13
STO 1	028- 44 1	g ABS	057- 43 16

TASTER	LYSPANEL	TASTER	LYSPANEL
RCL 3	058– 45 3	g ABS	064– 43 16
$x \geqslant y$	059– 34	f $x \leqslant y$	065– 42 10
f $x > y$	060– 42 20	g RTN	066– 43 32
GTO 1	061– 22 1	GTO 1	067– 22 1
RCL 4	062– 45 4	f LBL C	068–42,21,13
RCL 0	063– 45 0		

	REGISTRE		R_1: Løkketæller
R_0: δ_i	R_1: Foreløbig	R_2: Foreløbig	R_3: ϵ
R_4: $\triangle x$ grænse	R_5–R_8: Ubrugt		

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Hop til f LBL C		GTO C	
3	Skift til programmeringsstatus.		g P/R	068-42,21,13
4	Indtast underprogram til beregning af funktionen.	$f(x)$		
5	Skift til normalstatus.		g P/R	
6	Vælg USER-status.			
7	Initialisér grænser.		A	10^{-8}
8	(Valgfri) Indtast ε hvis De ikke vil	ϵ	R/S	ϵ
	ændre normalværdien for ε (10^{-8}),			
	men ønsker at gå videre			
	til TRIN 9:		R/S	10^{-8}
9	(Valgfri) Indtast $\triangle$ x.	$\triangle$ x-grænse	R/S	$\triangle$ x-grænse
10	Indtast startgæt og beregn			
	en rod.	x_0	B	x_i

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
11				
12	Fortsæt fra TRIN 8 ved indtastning af nye grænser.			
13	Ved beregninger med en ny funktion			
	f(×), hoppes til programmets afslutning		g RTN g BST	
	Skift til programmeringsstatus og slet			
	de linier, der definerer den gamle funk-		g P/R	
	tion (dvs. indtil f LBL C.		←...←	068-42,21,13
	Fortsæt fra TRIN 4.			

Eksempel:

Find den største rod i
$f(\chi) = \chi^6 - \chi - 1 = 0$
med $\chi_0 = 2$

Indtastning	Lyspanel	
GTO C		
g P/R	068-42,21,13	
STO 5		
6 y^x		
RCL 5 − 1 −	075- 30	
g P/R		
f FIX 9		
Vælg USER-status		
A	0.000000010	Normalværdier for ε og △
2 B	1.134724138	x_i
C	-0.000000004	$f(x_i)$

U-bådsjagt

De skal bruge Deres destroyer til at bestemme en fjendtlig U-båds position. Den gemmer sig i et område, der er delt op i 10×10 lige store, kvadratiske felter. Når det er gjort, skal De ødelægge den med en dybvandsbombe.

Når De indtaster et udgangstal (mellem 0 og 1) placerer HP-11C U båden midt i et eller andet af de 100 kvadrater (R;S), hvor R=Række og S=Søjle, og hvor R og S kan antage værdierne 0, 1, ..., 9.

De har et sonaranlæg til scanning af nettet. Indtast Deres egen (dvs. destroyerens) position (R;S) og tryk på [B]. Hvis U-båden ligger et af de 8 omgivende felter (eller lige under destroyeren), udlæses »1«. Ellers udlæses »0«.

Når De tror, at De har fanget U-båden, skal destroyeren føres hen i det felt, hvor U-båden formodes at gemme sig. Herefter kan De kaste en dybvandsbombe med [A]. Hvis De rammer U-båden, vil lyspanelet blinke »1«-taller som tegn på succes'en. Ellers giver den bare et »0« og flytter U-båden til et af de fire omgivende felter – i samme række eller søjle.

Spillet kan gøres mere spændende, hvis De trykker på [C] lige efter at have indlæst udgangstallet. Nu vil U-båden bevæge sig et felt efter hver sonarscanning. Den flytter sig altid i samme række eller søjle og bliver inden for 10×10-nettet.

En dybvandsbombe har en rækkevidde på 0,9. Når De kaster en bombe, behøver destroyeren ikke være midt i et felt. Så hvis De f.eks. kaster bomben fra (2,5;6,5), vil den ødelægge enhver U-båd i felterne (2;6), (2;7), (3;6) og (3;7).

Prøv om De kan sænke U-båden med kun én dybvandsbombe og uden at bruge sonaren mere end 10 gange. [D] kan på ethvert tidspunkt fortælle Dem, hvordan status er.

Statusformatet er XX, YY,
hvor XX = antallet af brugte dybvandsbomber
og YY = antallet af sonarscanninger.

TASTER	LYSPANEL	TASTER	LYSPANEL
[f] CLEAR [PRGM]	000–	[g] [RTN]	005– 43 32
[f] [LBL] [C]	001–42,21,13	[f] [LBL] [E]	006–42,21,15
1	002– 1	[f] CLEAR [REG]	007– 42 34
[STO] 0	003– 44 0	[g] [CF] 0	008–43, 5, 0
[g] [SF] 0	004–43, 4, 0	[STO] [RAN#]	009– 44 36

TASTER	LYSPANEL	TASTER	LYSPANEL
GSB 9	010– 32 9	f PSE	039– 42 31
STO 1	011– 44 1	g RTN	040– 43 32
GSB 9	012– 32 9	f LBL B	041–42,21,12
STO 2	013– 44 2	1	042– 1
f FIX 0	014–42, 7, 0	STO + 8	043–44,40, 8
g CLx	015– 43 35	R↓	044– 33
g RTN	016– 43 32	f FIX 0	045–42, 7, 0
f LBL A	017–42,21,11	g CF 1	046–43, 5, 1
1	018– 1	GSB 6	047– 32 6
STO + 7	019–44,40, 7	RCL 0	048– 45 0
R↓	020– 33	STO 5	049– 44 5
g SF 1	021–43, 4, 1	g F? 0	050–43, 6, 0
GSB 6	022– 32 6	GSB 5	051– 32 5
g $x\neq 0$	023– 43 30	RCL 3	052– 45 3
GTO 0	024– 22 0	g RTN	053– 43 32
1	025– 1	f LBL 5	054–42,21, 5
STO 5	026– 44 5	GSB 9	055– 32 9
GSB 5	027– 32 5	4	056– 4
g RTN	028– 43 32	$x\geq y$	057– 34
f LBL 0	029–42,21, 0	f $x>y$	058– 42 20
9	030– 9	GTO 0	059– 22 0
1/x	031– 15	RCL 5	060– 45 5
f FIX 3	032–42, 7, 3	CHS	061– 16
f PSE	033– 42 31	GTO 1	062– 22 1
f FIX 5	034–42, 7, 5	f LBL 0	063–42,21, 0
f PSE	035– 42 31	RCL 5	064– 45 5
f FIX 7	036–42, 7, 7	f LBL 1	065–42,21, 1
f PSE	037– 42 31	STO 6	066– 44 6
f FIX 9	038–42, 7, 9	GSB 9	067– 32 9

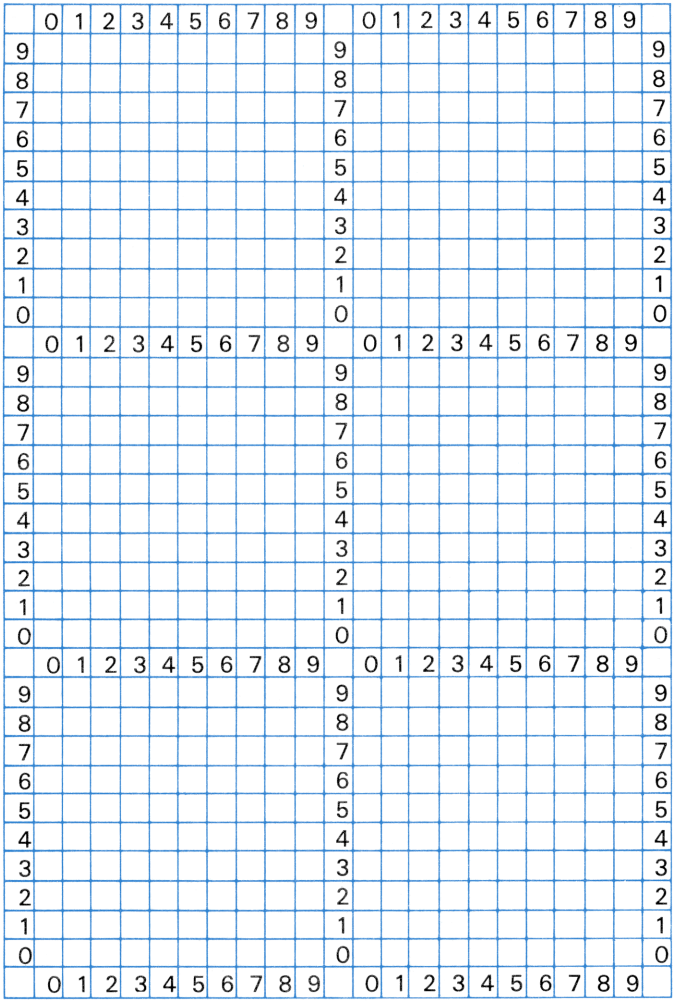
TASTER	LYSPANEL	TASTER	LYSPANEL
5	068– 5	f FIX 0	097–42, 7, 0
f $x \geq y$	069– 42 20	g RTN	098– 43 32
GTO 0	070– 22 0	f LBL 6	099–42,21, 6
RCL 1	071– 45 1	RCL 2	100– 45 2
GSB 1	072– 32 1	−	101– 30
STO 1	073– 44 1	$x \gtrless y$	102– 34
GTO 2	074– 22 2	RCL 1	103– 45 1
f LBL 0	075–42,21, 0	−	104– 30
RCL 2	076– 45 2	g →P	105– 43 26
GSB 1	077– 32 1	STO 4	106– 44 4
STO 2	078– 44 2	g F? 1	107–43, 6, 1
GTO 2	079– 22 2	GTO 0	108– 22 0
f LBL 1	080–42,21, 1	1	109– 1
RCL 6	081– 45 6	−	110– 30
+	082– 40	f LBL 0	111–42,21, 0
g $x<0$	083– 43 10	·	112– 48
GTO 0	084– 22 0	9	113– 9
9	085– 9	−	114– 30
$x \gtrless y$	086– 34	g $x<0$	115– 43 10
f $x \leq y$	087– 42 10	GTO 0	116– 22 0
g RTN	088– 43 32	0	117– 0
f LBL 0	089–42,21, 0	GTO 1	118– 22 1
RCL 6	090– 45 6	f LBL 0	119–42,21, 0
2	091– 2	1	120– 1
×	092– 20	f LBL 1	121–42,21, 1
−	093– 30	STO 3	122– 44 3
g RTN	094– 43 32	g RTN	123– 43 32
f LBL 2	095–42,21, 2	f LBL D	124–42,21,14
RCL 3	096– 45 3	f FIX 2	125–42, 7, 2

TASTER	LYSPANEL	TASTER	LYSPANEL
RCL 7	126– 45 7	f LBL 9	133–42,21, 9
RCL 8	127– 45 8	f RAN#	134– 42 36
EEX	128– 26	1	135– 1
2	129– 2	0	136– 0
÷	130– 10	×	137– 20
+	131– 40	g INT	138– 43 44
g RTN	132– 43 32	g RTN	139– 43 32

REGISTRE			R_1: Ubrugt
R_0: 0, 1	R_1: P_1	R_2: P_2	R_3: Svar
R_4: d	R_5: Bruges	R_6: Bruges	R_7: Bruges
R_8: Bruges			

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Vælg USER-status.			
3	Indtast udgangstal (mellem 0 og 1)	*n*	E	0
4	Gå til TRIN 6 for normalspil.			
5	Vælg specialspil (hvor U-båden			
	bevæger sig efter scanning).		C	1
6	SONAR	Række	ENTER	
	»0« når der ikke fås ekko	Søjle	B	0 eller 1
	»1« når der modtages ekko			
	eller			
	DYBVANDSBOMBE	Række	ENTER	
	»0« ved forbier	Søjle	A	0 eller
	Blinkende »1«-taller			blink
	viser, at U-båden er sænket.			

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
7	Gentag TRIN 6, indtil			
	U-båden er sænket.			
8	Udlæsning af status:		D	XX.YY
	XX = Antal affyrede dybvandsbomber			
	YY = Antal sonar scanninger.			
9	Nyt spil startes fra TRIN 3.			



Spilleplade til U-bådsjagten.

Eksempel 1:

Vælg USER-status

Indtastning	**Lyspanel**	
,58 [E]	***0,***	
Første træk:		
3[ENTER]8 [B]	***1,***	Ekko

De véd nu, at fjenden befinder sig i et af de felter, der er mærket med »X«.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagram over 1. træk

Andet træk:

4 [ENTER] 7 [B]	***0,***	Ikke noget ekko

Det betyder, at U-båden ikke kan være i de felter, der er mærket med (X)

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								(X)	(X)	X	4
3								(X)	(X)	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagram over 2. træk

Tredie træk:

2 ENTER 9 B *0,* Ikke noget ekko

Nu har De indkredset U-båden ganske udmærket. Den kan kun være i et af de to felter, der udelukkende er mærket med »×«.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								⊗	⊗	×	4
3								⊗	⊗	⊗	3
2								×	⊗	⊗	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Fjerde træk:

4 ENTER 9 B *1,* Ekko

Det eliminerer (2;7), så nu véd De med sikkerhed, hvor U-båden gemmer sig.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								⊗	⊗	×	4
3								⊗	⊗	⊗	3
2								⊗	⊗	⊗	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Femte træk:

4 ENTER 9 A 0,111
0,11111

0,1111111
0,111111111
0,111111111 U-båden blev ramt!

Eksempel 2:

Indtastning	**Lyspanel**
60 [E]	**0**
[C]	**1,**

Nu vil U-båden bevæge sig efter hver sonarscanning.

Første træk:

7 [ENTER] 4 [B] **1,** Ekko

U-båden gemte sig i et af de felter, der er mærket med »×« i venstre halvdel. Men U-båden bevæger sig efter scanningen, så nu kan den være i ethvert af de felter, der er mærket med »×« i diagrammets højre halvdel.

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9											9				X	X	X					9
8				X	X	X					8			X	X	X	X	X				8
7				X	X	X					7			X	X	X	X	X				7
6				X	X	X					6			X	X	X	X	X				6
5											5				X	X	X					5
4											4											4
3											3											3
2											2											2
1											1											1
0											0											0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

Diagram over 1. træk

Andet træk:

6 [ENTER] 4 [B] **0,** Ikke noget ekko.

Hermed har De elimineret en række felter (dem der er mærket med (x) i diagrammets venstre side), men U-båden har bevæget sig, så den nu kan være i de felter, der i højre side er mærket »×«.

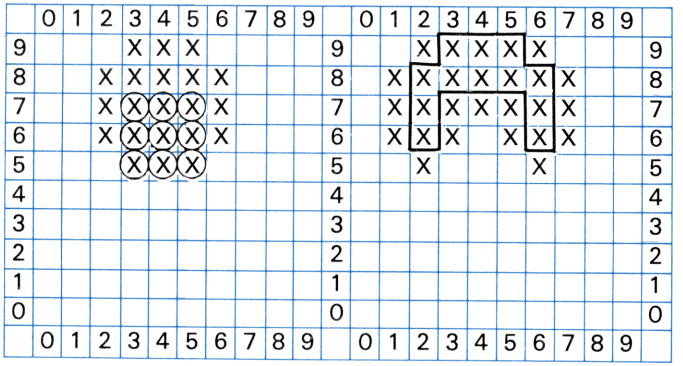


Diagram over 2. træk

Tredie træk:

7 ENTER 3 B **1,** Ekko

Dette træk udelukkede en mængde felter (venstre side), men stadig kommer ny muligheder til (højre side).

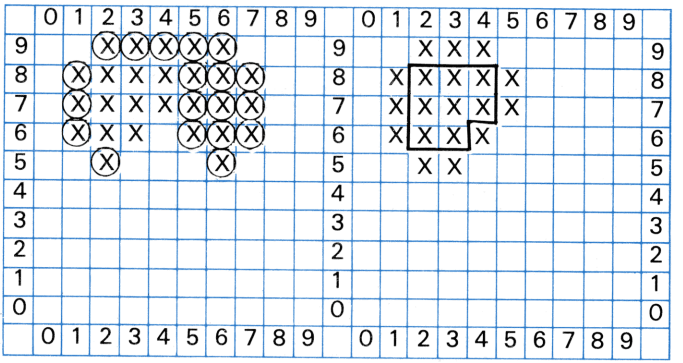


Diagram over 3. træk

Fjerde træk: Prøv med en dybvandsbombe...

8 ENTER 3 A

0,111
0,11111
0,1111111
0,111111111
0,111111111

Nogle gange er heldet bedre end forstanden: Der var en gang en U-båd her.

	0	1	2	3	4	5	6	7	8	9	
9											9
8				X							8
7											7
6											6
5											5
4											4
3											3
2											2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Finansregning
Annuiteter og rentes rente

Dette program kan bruges til at løse en lang række problemer, der vedrører samspillet mellem tid og penge. Følgende størrelser eller parametre kan alle bruges som inddata, og med undtagelse af rentesatsen kan programmet beregne enhver parameter, når to/tre af de andre er givet:

n: Antallet af beregningsterminer. F.eks. vil et 10-årigt lån med månedlige ydelser have 12×10=120 beretningsterminer.

i: Rentesatsen pr. termin. Rentesatsen opgives altid i procent. Hvis rentesatsen er 10% p.a., vil rentesatsen pr. måned være 10/12% eller ca. 0,83%.

PV: Hovedstolen, dvs. det oprindelige beløb.

PMT: Fast ydelse, der forfalder ved hver termin. Består af en rente- og en afdragsdel.

FV: Slutværdien af en opsparing eller restgælden for et lån.

Desuden kan programmet bruges til at bestemme den rente, der over et antal terminer er betalt på et lån samt lånets restgæld.

Programmet kan arbejde med såvel forudbetalte som efterbetalte ydelser. Efterbetalte ydelser forfalder ved afslutningen af en termin og er den almindeligste låneform, som f.eks. bruges ved almindelige sælgerpantebreve. En forudbetalt ydelse forfalder ved terminens start, og bruges stort set kun i forbindelse med leasingkontrakter.

Man vælger efterbetalt annuitet ved at trykke [A] og dernæst på [R/S], indtil »1« udlæses. Forudbetalt annuitet vælges ved at trykke [A] og [R/S], indtil »0« udlæses.

Programmet følger *fortegnskonventionen:* Udgifter er *negative,* hvorimod indtægter er *positive.* Det skulle være til at huske.

Ved at trykke [f] **CLEAR** [REG] nulstilles (slettes) alle registre således, at HP-11C er klar til løsning af en ny opgave. Det er dog en meget drastisk initialisering, og den er da heller ikke nødvendig. Man skal blot sørge for *manuelt* at nulstille de registre, der svarer til parametre, der ikke indgår i beregningen. F.eks. skal R_4, der indeholder *PMT,* nulstilles ved beregning af rentes rente.

Den følgende tabel giver en oversigt over de forskellige muligheder.

Oversigt over
beregninger af rentes rente og annuiteter
når rentesatsen er givet

Kombination af parametre	Anvendelser		Initialisering
	Efterbetalt	Forudbetalt	
N, PV, PMT (Indtast to af værdierne og beregn den sidste)	Almindelige lån Kursværdi for pantebreve med forfald udløb	Leasing	Brug [f] CLEAR [REG] eller nulstil R_5 [FV]
n, PV, PMT og *FV* Indtast tre værdier og beregn den fjerde)	Pantebreve med forfald før udløb	Leasing med scrap-værdi	Ingen
n, PMT, FV (Indtast to værdier og beregn den tredie)	Opsparing	Pensions-Kontrakt	Brug [f] CLEAR [REG] eller nulstil R_3 [PV]
n, PV, FV (Indtast to værdier og beregn den tredie)	Rentes rente, hvor det ikke har mening af skelne mellem forud- og efterbetalte ydelser		Brug [f] CLEAR [REG] eller nulstil R_4 [PMT]

Formel: $$-PV = \frac{PMT}{i} A\,[1-(1+i)^{-n}] + (FV)\,(1+i)^{-n}$$

Hvor A er 1 for efterbetalte annuiteter og $(1+i)$ for forudbetalte.

Bemærk:

- Hvis rentesatsen sættes til 0, gives fejlmeddelelsen ***Error 0.***
- Hvis n eller i er ekstremt store ($>10^6$) eller små ($<10^{-6}$), kan der blive beregnet forkerte resultater.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000–	STO 1	029– 44 1
f LBL A	001–42,21,11	g RTN	030– 43 32
f FIX 2	002–42, 7, 2	f LBL C	031–42,21,13
STO 2	003– 44 2	STO 3	032– 44 3
f LBL 4	004–42,21, 4	R/S	033– 31
R/S	005– 31	GSB 1	034– 32 1
0	006– 0	GSB 2	035– 32 2
g F? 0	007–43, 6, 0	CHS	036– 16
1	008– 1	STO 3	037– 44 3
g CF 0	009–43, 5, 0	g RTN	038– 43 32
g $x=0$	010– 43 40	f LBL D	039–42,21,14
g SF 0	011–43, 4, 0	STO 4	040– 44 4
GTO 4	012– 22 4	R/S	041– 31
f LBL B	013–42,21,12	1	042– 1
STO 1	014– 44 1	STO 4	043– 44 4
R/S	015– 31	GSB 1	044– 32 1
GSB 1	016– 32 1	$1/x$	045– 15
RCL 5	017– 45 5	RCL 3	046– 45 3
g LSTx	018– 43 36	GSB 2	047– 32 2
−	019– 30	×	048– 20
RCL 3	020– 45 3	CHS	049– 16
g LSTx	021– 43 36	STO 4	050– 44 4
+	022– 40	g RTN	051– 43 32
÷	023– 10	f LBL E	052–42,21,15
CHS	024– 16	STO 5	053– 44 5
g LN	025– 43 12	R/S	054– 31
RCL 6	026– 45 6	GSB 1	055– 32 1
g LN	027– 43 12	RCL 3	056– 45 3
÷	028– 10	+	057– 40

TASTER	LYSPANEL	TASTER	LYSPANEL
RCL 7	058– 45 7	y^x	076– 14
÷	059– 10	STO 7	077– 44 7
CHS	060– 16	1	078– 1
STO 5	061– 44 5	$x \gtrless y$	079– 34
g RTN	062– 43 32	−	080– 30
f LBL 1	063–42,21, 1	RCL 4	081– 45 4
1	064– 1	RCL 8	082– 45 8
RCL 2	065– 45 2	÷	083– 10
g %	066– 43 14	RCL 0	084– 45 0
STO 8	067– 44 8	×	085– 20
1	068– 1	×	086– 20
STO 0	069– 44 0	g RTN	087– 43 32
+	070– 40	f LBL 2	088–42,21, 2
STO 6	071– 44 6	RCL 5	089– 45 5
g F? 0	072–43, 6, 0	RCL 7	090– 45 7
STO 0	073– 44 0	×	091– 20
RCL 1	074– 45 1	+	092– 40
CHS	075– 16		

REGISTRE			R_I: Ubrugt
R_0: i eller $1+i$	R_1: n	R_2: $i(\%)$	R_3: PV
R_4: PMT	R_5: FV	R_6: $1+i$	R_7: $(1+i)^{-n}$
R_8: $i/100$	R_9—$R_{.4}$: Ubrugt		

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Vælg USER-status.			
3	Initialisering.		f CLEAR REG	
4	Indtast rentesatsen	*i*(%)	A	*i*(%)
5	Vælg mellem efterbetalt		R/S	1.00
	(1) og forudbetalt (0)		R/S	0.00
	annuitet.			
6	Indtast kendte værdier:			
	Antal terminer	*n*	B	*n*
	Hovedstol	*PV*	C	*PV*
	Fast ydelse	*PMT*	D	*PMT*
	Slutværdi	*FV*	E	*FV*
7	Beregn den ukendte størrelse:			
	Antal terminer		B R/S	*n*
	Hovedstol		C R/S	*PV*
	Fast ydelse		D R/S	*PMT*
	Slutværdi		E R/S	*FV*
8	Ny opgave udføres ved at starte igen			
	fra TRIN 4 og modificere de lagrede			
	værdier. Parametre, der ikke			
	indgår i en opgave, skal sættes til 0.			
9	Ved at fortsætte fra TRIN 3 kan			
	der startes på en ny opgave,			
	hvor ingen parametre overføres.			

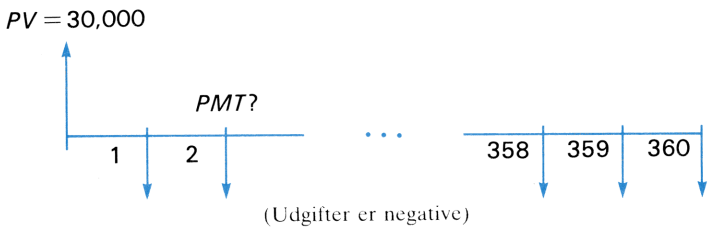
Eksempel 1:

Hvis der indsættes 155 kr. på en bankbog til 5,75% rente p.a. med månedlig rentetilskrivning, hvor meget står der da på kontoen efter 9 år?

Indtastning	Lyspanel	
[f] CLEAR [REG]		
Vælg USER-status.		
5,75 [ENTER] 12 [÷] [A]	***0,48***	Rentesats (i % pr. måned).
[R/S]	***0,00***	
[R/S]	***1,00***	Efterbetalt annuitet.*
9 [ENTER] 12 [×] [B]	***108,00***	Antal terminer.
155 [CHS] [C]	***-155,00***	Hovedstol.
[E] [R/S]	***259,74***	Slutværdi efter 9 år.

Eksempel 2:

Et lån på 30.000,00 kr. skal forrentes og afdrages med lige store månedlige ydelser over 30 år. Rentesatsen er 13% p.a. Hvor stor er den månedlige ydelse?

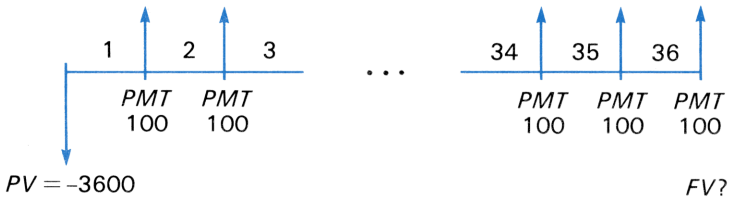


*) I denne opgave er der ikke behov for at vælge mellem forudbetalt og efterbetalt annuitet, men det skal gøres, og vi vil bruge efterbetalt annuitet i næste eksempel.

Indtastning	Lyspanel	
f CLEAR REG		
13 ENTER 12 ÷ A	***1,08***	*i* (i % pr. måned)
30 ENTER 12 × B	***360,00***	*n*
30000 C	***30.000,00***	*PV*
D R/S	***-331,86***	*PMT*

Eksempel 3:

I forbindelse med et forbrugskøb oprettes et pantebrev på 3.600,00 kr., som skal forrentes og afdrages over 36 måneder. Långiver (der er i familie med låntager) accepterer en rentesats på 10% p.a. og en fast månedlig ydelse på 100 kr. Til gengæld forfalder restgælden i forbindelse med den 36. termin. Hvor stor er denne restgæld?
(Bemærk, at diagrammet beskriver forløbet set fra *långivers* synspunkt).



Indtastning	Lyspanel	
f CLEAR REG		
10 ENTER 12 ÷ A	***0,83***	*i* (i % pr. måned)
36 B	***36,00***	*n*
3600 CHS C	***-3.600,00***	*PV*
100 D	***100,00***	*PMT*
E R/S	***675,27***	*FV*

Bemærk, at den 36. ydelse bliver kr. 675,27 + 100,00, eller ialt kr. 765,27.

Eksempel 4:

Programmet kan også bruges til at bestemme, hvor megen rente der efter et vist antal terminer er betalt på et lån, og hvor stor restgælden er efter en bestemt termin. Den betalte rente mellem to terminer n_1 og n_2 svarer til summen af ydelserne minus den del af disse, der er brugt til at ned-

bringe gælden (afdragsdelen). Afdragsdelen er lig med restgælden efter termin n_2 minus restgælden efter termin n_1.

Der er givet et 30-årigt lån på 50.000,00 kr. Det forrentes og afdrages med månedlige ydelser, og rentesatsen er 14% p.a. Find restgælden efter 24. termin og bestem, hvor megen rente der er betalt i terminerne 13-24. (Dvs. mellem 12. og 24. termin!)

Først må vi beregne den fast ydelse:

Indtastning	**Lyspanel**	
f CLEAR REG		
360 B	***360,00***	n = 30 år
14 ENTER 12 ÷ A	***1,17***	i
50000 CHS C	***-50.000,00***	PV
D R/S	***592,44***	PMT

Efter 24 terminer er restgælden:

24 B E R/S	***49.749,56***	Restgæld efter 24. termin

Denne restgæld lagres, og restgælden efter 12. termin bestemmes:

STO 9	***49.749,56***	
12 B E R/S	***49.883,48***	Restgæld efter 12. termin

Dvs. for terminerne 13-24 udgør afdragsdelen:

RCL 9 −	***133,92***	Afdrag i terminerne 13-24

For terminerne 13-24 udgør den betalte rente derfor:

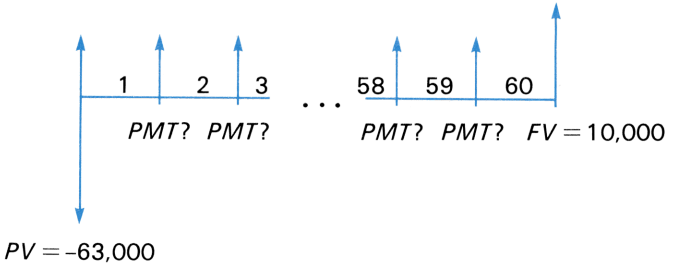
RCL 4	***592,44***	PMT
12 ×	***7.109,23***	De tolv ydelser i terminerne 13-24
$x \gtrless y$ −	***6.975,31***	Rentedelen

Eksempel 5:

Et leasingfirma påtænker at anskaffe en borddatamat til 63.000,00 kr. Firmaet forventer at opnå en 13% forrentning af den investerede kapital ved at lade en klient lease maskinen over en 5-årig periode. Leasingfirmaet bevarer ejendomsretten til borddatamaten, som formodes at kunne sælges for 10.000,00 kr. efter leasingperiodens udløb.

Hvor stor en leasingafgift skal firmaet beregne sig for at opnå den ønskede forrentning?

(Leasingafgiften erlægges ved *starten* af hver beregningstermin, så vi skal nu skifte til arbejde med forudbetalte annuiteter.)



Indtastning	Lyspanel	
f CLEAR REG		
13 ENTER 12 ÷ A	***1,08***	*i*
R/S	***0,00***	Forudbetalt annuitet
5 ENTER 12 × B	***60,00***	*n*
63000 CHS C	***-63.000,00***	*PV*
10000 E	***10.000,00***	*FV* (scrap-værdi)
D R/S	***1.300,16***	*PMT*

Hvis prisen var 70.000,00 kr., hvor stor skulle leasingafgiften da være?

70000 CHS C	***-70.000,00***	
D R/S	***1.457,73***	*PMT*

t-test

t-test for parvise observationer

Der er givet et sæt parvise observationer fra to normalfordelte populationer med ukendte middelværdier μ_1 og μ_2.

x_i	x_1	$x_2 \dots x_n$
y_i	y_1	$y_2 \dots y_n$

Idet differenserne betragtes som statistiske variable fås:

$$D_i = x_i - y_i$$

$$\bar{D} = \frac{1}{n} \sum_{i=1}^{n} D_i$$

$$s_D = \sqrt{\frac{\Sigma D_i^2 - \frac{1}{n}(\Sigma D_i)^2}{n-1}}$$

Testværdien

$$t = \frac{\bar{D}}{s_D}\sqrt{n}$$

som har n-1 frihedsgrader, kan bruges til at teste nulhypotesen $H_0 : \mu_1 = \mu_2$.

t-test for to middelværdier

Antag at $(x_1, x_2, \ldots, x_{n1})$ og $(y_1, y_2 \ldots, y_{n2})$ er to uafhængige stikprøver fra to normalfordelte populationer, der har ukendte middelværdier μ_1 og μ_2 og samme ukendte varians σ^2.

Vi ønsker at teste nulhypotesen H_0: $\mu_1 - \mu_2 = d$.

Formler:

$$\bar{x} = \frac{1}{n_1}\sum_{i=1}^{n_1} x_i$$

$$\bar{y} = \frac{1}{n_2}\sum_{i=1}^{n_2} y_i$$

$$t = \frac{\bar{x} - \bar{y} - d}{\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}\sqrt{\frac{\Sigma x_i^2 - n_i\bar{x}^2 + \Sigma y_i^2 - n_2\bar{y}^2}{n_1 + n_2 - 2}}}$$

Vi kan bruge testværdien, der er t-fordelt med n_1+n_2-2 frihedsgrader til at teste nulhypotesen H_0.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000–	STO · 2	029– 44 .2
f LBL A	001–42,21,11	RCL 0	030– 45 0
−	002– 30	STO · 0	031– 44 .0
Σ+	003– 49	0	032– 0
R/S	004– 31	f CLEAR Σ	033– 42 32
f LBL C	005–42,21,13	R/S	034– 31
g $\bar{x}$	006– 43 0	f LBL D	035–42,21,14
STO 6	007– 44 6	STO 6	036– 44 6
R/S	008– 31	RCL · 1	037– 45 .1
g s	009– 43 48	RCL · 2	038– 45 .2
R/S	010– 31	STO 4	039– 44 4
RCL 6	011– 45 6	$x \gtrless y$	040– 34
RCL 0	012– 45 0	STO 3	041– 44 3
$\sqrt{x}$	013– 11	g $\bar{x}$	042– 43 0
×	014– 20	STO 8	043– 44 8
$x \gtrless y$	015– 34	$x \gtrless y$	044– 34
÷	016– 10	RCL 0	045– 45 0
R/S	017– 31	×	046– 20
RCL 0	018– 45 0	RCL · 0	047– 45 .0
1	019– 1	÷	048– 10
−	020– 30	STO 7	049– 44 7
R/S	021– 31	$x \gtrless y$	050– 34
GTO C	022– 22 13	−	051– 30
f LBL B	023–42,21,12	RCL 6	052– 45 6
Σ+	024– 49	−	053– 30
R/S	025– 31	RCL 4	054– 45 4
RCL 1	026– 45 1	RCL 3	055– 45 3
STO · 1	027– 44 .1	RCL 7	056– 45 7
RCL 2	028– 45 2	×	057– 20

TASTER	LYSPANEL	TASTER	LYSPANEL
[−]	058– 30	[√x]	072– 11
[RCL] 2	059– 45 2	[RCL] 0	073– 45 0
[+]	060– 40	[1/x]	074– 15
[RCL] 1	061– 45 1	[RCL] [·] 0	075– 45 .0
[RCL] 8	062– 45 8	[1/x]	076– 15
[×]	063– 20	[+]	077– 40
[−]	064– 30	[√x]	078– 11
[RCL] 0	065– 45 0	[×]	079– 20
[RCL] [·] 0	066– 45 .0	[÷]	080– 10
[+]	067– 40	[R/S]	081– 31
2	068– 2	[RCL] 9	082– 45 9
[−]	069– 30	[R/S]	083– 31
[STO] 9	070– 44 9	[GTO] [D]	084– 456
[÷]	071– 10	[RCL] 6	085– 22 14

REGISTRE*			
			R_I: Ubrugt
R: n , n ; n	R: Σx, Σy; ΣD	R: Σx^2, Σy^2; ΣD^2	R_3: Σx;
R_4: Σx^2;	R_5: Ubrugt	R_6: d; $\bar{D}$	R_7: $\bar{x}$;
R_8: $\bar{y}$;	R_9: df;	R_0: n_1;	R_1: Σx;
R_2: Σx^2;	R_3-R_4: Ubrugt		

* værdier foran semikolon bruges ved t-test for to middelværdier. Værdier efter bruges ved parvise observationer.

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
	t-test for parvise observationer.			
1	Indtast programmet.			
2	Initialisering af programmet.		[f] CLEAR [REG]	
3	Vælg USER-status.			

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
4	Indtast x_i.	x_i	ENTER	
	Indtast y_i.	y_i	A	i
5	Korrigér evt. fejl.			
	(Gentag TRIN 4-5 for alle n par.)	x_k	ENTER	
		y_k	− g Σ−	k–1
6	Beregning af testværdier:			
	D		C	$\bar{D}$
	s_D		R/S	s_D
	t		R/S	t
	df (antal frihedsgrader).		R/S	df
7	Resultaterne kan genudlæses ved at gentage TRIN 6.			
	t-test for to middelværdier.			
1	Indtast programmet (hvis det ikke allerede er lagret).			
2	Initialisering.		f CLEAR REG	
3	Vælg USER-status.			
4	Indtast x_i.	x_i	B	i
5	Korrigér evt. fejl. TRIN 4-5	x_k	g Σ−	k–1
	gentages for alle n_1 værdier.			
6	Initialisering før indlæsning af		R/S	
	det andet datasæt.			
7	Indtast y_i.	y_i	B	i
8	Korrigér evt. fejl. TRIN 7-8	y_k	g Σ−	k–1
	gentages for alle n_2 værdier.			
9	Beregn testværdier for parameteren d:			
	t	d	D	t
	df (antal frihedsgrader).		R/S	df
10	Resultaterne kan genudlæses			
	ved at gentage TRIN 9.			

Eksempel 1:

x_i	14	17.5	17	17.5	15.4
y_i	17	20.7	21.6	20.9	17.2

Find $\bar{D}$, s_D, t og df for dette sæt parvise observationer fra normalfordelte populationer med ukendte middelværdier.

Indtastning	**Lyspanel**	
f FIX 4		
Vælg USER-status		
f CLEAR REG		
14 ENTER 17 A	**1,0000**	
17 ENTER 15 A	**2,0000**	(Fejl)
17 ENTER		
15 − g Σ−	**1,0000**	(Korrektion)
17.5 ENTER 20.7 A	**2,0000**	
17 ENTER 21.6 A	**3,0000**	
17.5 ENTER 20.9 A	**4,0000**	
15.4 ENTER 17.2 A	**5,0000**	
C	**−3,2000**	$\bar{D}$
R/S	**1,0000**	s_D
R/S	**−7,1554**	t
R/S	**4,0000**	df

Eksempel 2:

x	79	84	108	114	120	103	122	120		
y	91	103	90	113	108	87	100	80	99	54

Find t og df for ovenstående data, idet d=0 og nulhypotesen således er H_0: $\mu_1 = \mu_2$.

Indtastning	**Lyspanel**
f CLEAR REG	
79 B 84 B 99	
B 99 g Σ−	**2,0000**
108 B 114 B	
120 B 103 B	
122 B 120 B	**8,0000**
R/S	**0,0000**
91 B 103 B	

Indtastning	Lyspanel	
90 [B] 113 [B]		
108 [B] 87 [B]		
100 [B] 80 [B]		
99 [B] 54 [B]	**10,0000**	
0 [D]	**1,7316**	*t*
[R/S]	**16,0000**	*df*

X^2-test

Programmet kan bruges til χ^2-evaluering af forskellen mellem et sæt observerede og de tilsvarende forventede værdier, idet man bruger

$$\chi_i^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \qquad (df = n - 1)$$

hvor O_i = den observerede værdi
og E_i = forventningsværdien

Hvis alle forventningsværdier er ens gælder:

$E = E_i = \Sigma O_i/n$ for alle i

hvoraf $$\chi_2^2 = \frac{n\Sigma O_i^2}{\Sigma O_i} - \Sigma O_i.$$

Bemærkning: Hvis data falder i grupper (intervaller), bør man vælge disse således, at ingen gruppe har en forventningsværdi på mindre end 5. Eventuelt må man slå grupper sammen. Hvis man f.eks. skal afgøre om et forelagt datasæt er normalfordelt, kan det være nødvendigt at slå ydergrupper sammen.

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000–	f LBL B	022–42,21,12
f LBL A	001–42,21,11	Σ+	023– 49
ENTER	002– 36	R/S	024– 31
R↓	003– 33	f LBL D	025–42,21,14
−	004– 30	RCL 0	026– 45 0
g x^2	005– 43 11	RCL 0	027– 45 0
g R↑	006– 43 33	RCL 2	028– 45 2
÷	007– 10	×	029– 20
g F? 0	008–43, 6, 0	RCL 1	030– 45 1
CHS	009– 16	÷	031– 10
STO + 1	010–44,40, 1	RCL 1	032– 45 1
1	011– 1	−	033– 30
g F? 0	012–43, 6, 0	R/S	034– 31
CHS	013– 16	RCL 1	035– 45 1
STO + 0	014–44,40, 0	RCL 0	036– 45 0
RCL 0	015– 45 0	÷	037– 10
g CF 0	016–43, 5, 0	R/S	038– 31
R/S	017– 31	GTO D	039– 22 14
f LBL C	018–42,21,13	f LBL E	040–42,21,15
RCL 1	019– 45 1	g SF 0	041–43, 4, 0
R/S	020– 31	GTO A	042– 22 11
GTO C	021– 22 13		

REGISTRE			R_1: Ubrugt
R_0: Indeks	R_1: χ_2^2, ΣO_i	R_2: ΣO_i^2	R_3: Bruges
R_4: Bruges	R_5: Bruges	R_6—R_9:	

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
	Forskellige forventningsværdier.			
1	Indtast programmet.			
2	Initialisering.		f CLEAR REG	
3	Vælg USER-status.			
4	Indtast O_i.		ENTER	
	Indtast E_i.	O_i	A	i
5	Korrigér evt. fejl: Gentag	E_i	ENTER	
	TRIN 4-5 for alle n værdier.	O_k	E	$k-1$
6	Beregn χ^2-værdien.	E_k	C	χ_1^2
7	Værdien kan genudlæses			
	ved at gentage TRIN 6.			
	Ens forventningsværdier.			
	(E er middeltallet for de			
	observerede værdier).			
1	Indtast programmet			
	(hvis det ikke allerede er lagret).			
2	Initialisering.		f CLEAR REG	
3	Vælg USER-status.			
4	Indtast O_i.	O_i	B	i
5	Korrigér evt. fejl.	O_h	g Σ-	$h-1$
6	Beregn χ^2.		D	χ_2^2
	Beregn E.		R/S	E
7	Resultaterne kan genudlæses ved at			
	gentage TRIN 6.			

* Indtast g CF 0 efter trin 2.

Eksempel 1:

Brug χ^2-test til at vurdere, om de observerede data, O_i, har samme fordeling som forventningsværdierne, E_i.

O_i	8	50	47	56	5	14
E_i	9.6	46.75	51.85	54.4	8.25	9.15

Indtastning	**Lyspanel**	
[f] CLEAR [REG] [f] [FIX] 4		
[g] [CF] 0		
Vælg USER-status		
8 [ENTER] 9.6 [A]	**1,0000**	
50 [ENTER] 46.75 [A]	**2,0000**	
47 [ENTER] 51.85 [A]	**3,0000**	
56 [ENTER] 54.4 [A]	**4,0000**	
100 [ENTER] [A]	**5,0000**	(Fejl)
100 [ENTER] [E]	**4,0000**	(Korrektion)
5 [ENTER] 8.25 [A]	**5,0000**	
14 [ENTER] 9.15 [A]	**6,0000**	
[C]	**4,8444**	χ_1^2

Eksempel 2:

Den følgende tabel opregner resultatet af et forsøg, hvor en sædvanlig terning blev kastet 120 gange. χ^2 bruges til at afgøre, om terningen er ærlig. Antallet af frihedsgrader er 5, og forventningsværdierne er ens for alle grupper.

Antal øjne (i)	1	2	3	4	5	6
Antal udfald (O_i)	25	17	15	23	24	16

Indtastning	**Lyspanel**	
[f] CLEAR [REG]		
25 [B] 17 [B] 19 [B]	**3,0000**	(Fejl)
19 [g] [$\Sigma-$]	**2,0000**	(Korrektion)
15 [B] 23 [B] 24 [B]		
16 [B]	**6,0000**	
[D]	**5,0000**	χ_2^2
[R/S]	**20,0000**	E

Nulhypotesen er, at der ikke er forskel på fordelingerne af de observerede data og forventningsværdierne. For 5 frihedsgrader vil en χ^2-værdi på 5,00 svare til mindre end 5% sandsynlighed for en forskel i de to fordelinger. Først ved 11,07 vil der være 95% sandsynlighed for en forskel. Så der ikke er noget, der taler imod terningens ærlighed.

Numerisk integration (diskrete tilfælde)

Programmet kan bruges til at integrere en funktion med kendt funktionsværdi i en række ækvidistante punkter. Integralet kan tilnærmes ved hjælp af trapezmetoden eller Simpson's metode.

Formler:
Lad $x_0, x_1, \ldots, x_n$ være n+1 ækvidistante punkter ($x_j = x_0 + hj$, j = 0, 1, 2, ..., n), i hvilke funktionsværdierne $f(x_0), f(x_1), \ldots, f(x_n)$ er kendt.

Integralet: $\int_{x_0}^{x_n} f(x)dx$ kan tilnærmes ved hjælp af:

1. Trapezmetoden.

$$\int_{x_0}^{x_n} f(x)dx \sim \frac{h}{2}\left[f(x_0) + 2\left(\sum_{j=1}^{n-1} f(x_j)\right) + f(x_n)\right]$$

2. Simpson's metode.

$$\int_{x_0}^{x_n} f(x)dx \sim \frac{h}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + \ldots 4f(x_{n-3})\right.$$

$$\left. + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right]$$

Simpson's metode kræver, at n er lige. Hvis n er ulige, udlæses **Error 0.**

TASTER	LYSPANEL	TASTER	LYSPANEL
f CLEAR PRGM	000–	STO 0	005– 44 0
f LBL A	001–42,21,11	STO 5	006– 44 5
STO 4	002– 44 4	0	007– 0
R/S	003– 31	STO 3	008– 44 3
f LBL B	004–42,21,12	f LBL 1	009–42,21, 1

TASTER	LYSPANEL	TASTER	LYSPANEL
R/S	010– 31	GSB 3	035– 32 3
f LBL B	011–42,21,12	3	036– 3
STO 1	012– 44 1	RCL 5	037– 45 5
GSB 2	013– 32 2	f LBL 0	038–42,21, 0
ENTER	014– 36	RCL 1	039– 45 1
+	015– 40	−	040– 30
STO + 5	016–44,40, 5	RCL 4	041– 45 4
1	017– 1	×	042– 20
STO + 3	018–44,40, 3	$x \gtrless y$	043– 34
RCL 3	019– 45 3	÷	044– 10
R/S	020– 31	R/S	045– 31
f LBL B	021–42,21,12	f LBL 2	046–42,21, 2
STO 1	022– 44 1	ENTER	047– 36
GSB 2	023– 32 2	+	048– 40
STO + 5	024–44,40, 5	STO + 0	049–44,40, 0
1	025– 1	g RTN	050– 43 32
STO + 3	026–44,40, 3	f LBL 3	051–42,21, 3
RCL 3	027– 45 3	2	052– 2
GTO 1	028– 22 1	÷	053– 10
f LBL C	029–42,21,13	f FRAC	054– 42 44
2	030– 2	g $x=0$	055– 43 40
RCL 0	031– 45 0	g RTN	056– 43 32
GTO 0	032– 22 0	0	057– 0
f LBL D	033–42,21,14	÷	058– 10
RCL 3	034– 45 3	g RTN	059– 43 32

	REGISTRE		R_i: Ubrugt
R_0: Bruges	R_1: $f(x_i)$	R_2:	R_3: n
R_4: h	R_5: Bruges	R_6–R_9: Ubrugt	

TRIN	INSTRUKTION	INDDATA	TASTER	UDDATA
1	Indtast programmet.			
2	Vælg USER-status.			
3	Indtast intervallet mellem x-værdierne, h.	h	[A]	h
4	Indtast funktionsværdien			
	f(x_j) for x_j.	$f(x_j)$	[B]	j
	Gentag TRIN 4 for alle n+1 værdier.			
5	Beregn integralet ved			
	hjælp af trapezmetoden		[C]	Trap $\int$
	eller ved hjælp af			
	Simpson's metode.			
	(n skal være lige.)		[D]	Simp $\int$
6	Start fra TRIN 3 ved ny beregning.			

Eksempel:

Find en tilnærmelse til integralet 1, hvor funktionen f(x) er givet af værdierne f(x_i) opført i nedenstående tabel, idet såvel trapezmetoden som Simpson's metode anvendes:

$$\int_0^2 f(x)dx$$

i	0	1	2	3	4	5	6	7	8
x_i	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
$f(x_i)$	2	2.8	3.8	5.2	7	9.2	12.1	15.6	20

Indtastning	Lyspanel	
f FIX 2		
Vælg USER-status		
.25 A	**0,25**	
2 B	**0,00**	
2.8 B	**1,00**	
3.8 B	**2,00**	
5.2 B	**3,00**	
7 B	**4,00**	
9.2 B	**5,00**	
12.1 B	**6,00**	
15.6 B	**7,00**	
20 B	**8,00**	
C	**16,68**	Trapezmetoden.
D	**16,56**	Simpson's metode.

Afsnit 2

Programmeringsteknik

Struktur

HP-11C's sprog er kunstsprog i modsætning til f.eks. dansk, der er et naturligt sprog. Et naturligt sprogs struktur gør det forståeligt, selvom man kun kender en delmængde af det. Det samme gælder ikke umiddelbart for kunstsproget (programmeringssproget). Men De kan selv gøre uhyre meget for at give de programmer, som De selv skriver, en struktur, der gør dem lette at overskue, lette at editere og lette at kommunikere. I dette afsnit vil vi præsentere nogle programmeringsredskaber, der erfaringsmæssigt giver programmer, der er ikke blot er brugbare – men også brugervenlige.

Problemformulering

De fleste, der netop har lært at betjene en programmérbar maskine af en eller anden slags, begår samme kapitalbrøler: De tror, at programmeringen er det vigtigste. (Nogle kommer aldrig ud af denne vildfarelse.) I virkeligheden viser det sig, at problemformuleringen er alfa og (næsten) omega i enhver programmeret løsningsmetode; det er under problemformuleringen, man skal beslutte sig til, hvad programmet egentlig skal kunne klare, og hvordan det, i hovedtrækkene, skal gøres. Man skal altså først se at få orden på de overordnede mål. Det vil altid være nødvendigt med kompromisser – og det vil spare tid og bekymringer, hvis man med det samme får fastlagt kompromisserne. Det er ligesom, når man skal ud på en rejse: Først når man har fundet ud af, hvor man vil hen, og hvordan man kunne tænke sig at komme derhen, giver det mening at slå op i køreplanen.

Eksempel: Antag at vi skal finde rødderne til andengradsligningen $ax^2 + bx + c = 0$, hvor a, b og c er konstanter. Den simplest mulige problemformulering vil være: »Med konstanterne a, b og c som inddata ønsker vi at finde *begge* løsninger til ligningen $ax^2 + bx + c = 0$«.

Dette kan synes meget indlysende, og det kan forekomme temmelig overflødigt at gøre så megen stads ud af så lidt. Men vi har dog fået fastlagt to vigtige forhold: hvilke inddata, der kræves, og hvilke svar vi ønsker. Dette er i princippet nok til at gå videre til næste del af problemformuleringen: valg af algoritme (løsningsmetode).

Algoritmen

En algoritme er ikke et program, men en beskrivelse af de processer, der skal til for at løse et problem. En algoritme bør være niveaudelt, så man starter ret uspecifikt, og først senere udfylder den med detaljer. Algoritmevalget afhænger helt af problemformuleringen. I vort tilfælde er kravet til løsningerne, at *begge* løsninger skulle findes.
Kravet er ubetinget, så vi skal altså finde såvel reelle som komplekse løsninger. Det blev antydet, at problemformuleringen faktisk er meget nødtørftig; man burde have diskuteret, hvilken betydning der skal tillægges nøjagtighed, udførselshastighed og programstørrelse. Det er netop ved afvejningen af disse forhold, man bliver nødt til at finde brugbare kompromisser.
I dette tilfælde vælger vi den simpleste løsning, den som alle i forvejen kender:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

På basis af denne formel, kan vi opstille en algoritme:

Bestem $b^2 - 4ac$.

Hvis differensen er positiv, bestemmes

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Hvis differensen er negativ, bestemmes

$$\frac{\pm \sqrt{|b^2 - 4ac|}}{2a} \text{ samt } -b/2a$$

Når b^2-$4ac$ er positiv, vil der være to reelle løsninger. Når differensen er negativ, vil der være to komplekse løsninger.

Bemærk, at algoritmen ikke siger noget om, *hvordan* selve programmeringen eller beregningen skal finde sted. Den beskriver, *hvad* der skal gøres, og i hvilken rækkefølge det skal gøres.

Det er tydeligt, at algoritmen indeholder mønstre, der gentages. Ved udfyldning af algoritmen skal man holde sig sådanne gentagelser for øje.

De kan måske programmeres som pladsbesparende underprogrammer. Hvis vi anvender de tre hypotetiske dataregistre R_A, R_B og R_C, kan algoritmen nu udbygges således:

1. Der skiftes fortegn for *b*, og *-b* divideres med *2a*.
2. *-b*/*2a* lagres i R_A.
3. *b* kvadreres, og *4ac* trækkes fra kvadratet.
4. b^2-*4ac* lagres i R_B.
5. Kvadratroden uddrages af den numeriske værdi af tallet i R_B, og resultatet divideres med *2a*.
6. $\sqrt{|R_B|}/2a$ lagres i R_C.
7. Hvis tallet i R_B er positivt eller 0, lægges tallene i R_A og R_C sammen, og resultatet udlæses.
8. Hvis tallet i R_B er mindre end 0, udlæses tallene i R_A og R_C hver for sig (kompleks rod).
9. Fortegnet for tallet i R_C skiftes, og proceduren i 7 og 8 gentages én gang.

På dette tidspunkt er algoritmen udvidet til at tage hensyn til lagring af mellemresultater. Samtidig defineres de enkelte operationer bedre, og der tages klart hensyn til formatet af uddata.

Det er naturligvis op til den enkelte at bestemme, hvor raffineret en algoritme skal være. Men man skal altid gøre sig følgende klart:

Hvis algoritmen er grundigt gennemarbejdet, er der mindre risiko for, at den senere skal revideres.

Hvis algortimen er kompliceret, er den meget vanskelig at revidere.

Følgende algoritme er så tilpas gennemarbejdet, at den må siges at være acceptabel. Samtidig er den så tilpas overskuelig, at man kan ændre på den, hvis man senere skulle få en lys idé.

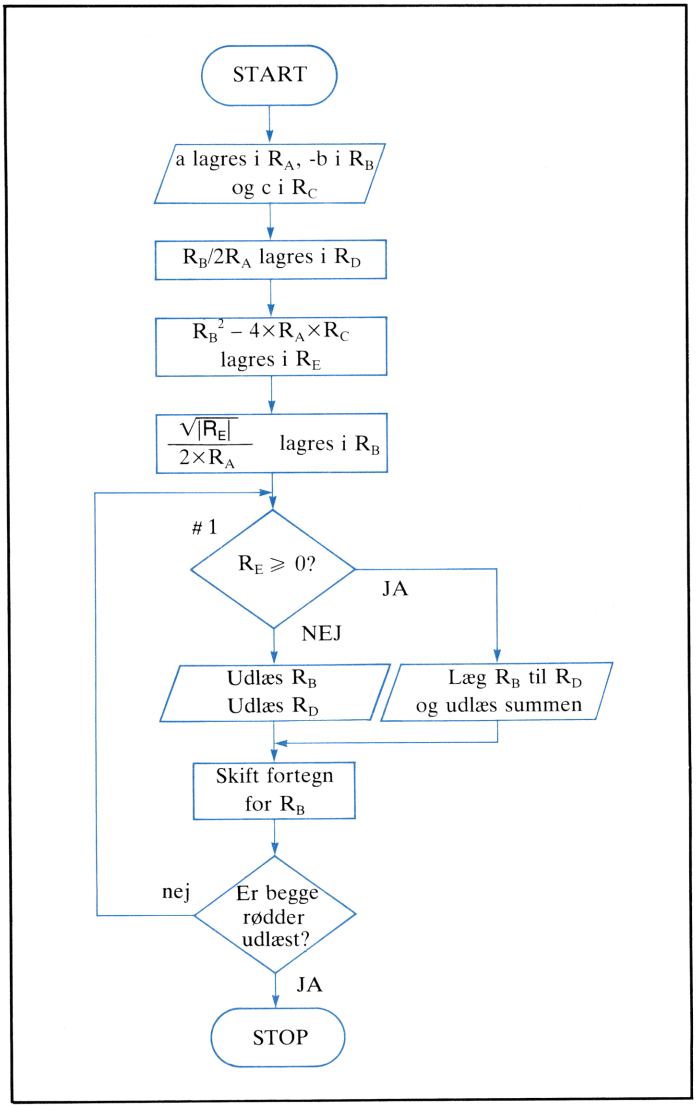
De hypotetiske registre R_A, R_B, R_C, R_D og R_E bruges i algoritmen.

1. *a* lagres i R_A, *-b* lagres i R_B, og *c* lagres i R_C.
2. R_B divideres med 2 gange R_A, og resultatet lagres i R_D.
3. R_B kvadreres, og herfra trækkes 4 gange R_A gange R_C, hvorefter resultatet lagres i R_E.
4. Kvadratroden uddrages af den numeriske værdi af R_E, og resultatet divideres med 2 gange R_A. Resultatet lagres i R_B.
5. Hvis R_E er positiv eller 0, lægges R_B til R_D, og resultatet udlæses.
6. hvis R_E er negativ, udlæses R_B og R_D hver for sig.
7. Skift fortegn for R_B og gentag 5 og 6 én gang.

Denne algoritme tager i betragtning, at tal skal lagres ved indlæsningen og genbruger desuden register R_B således, at det totale registerkrav formindskes.

Rutediagrammer

Algoritmen kan anskueliggøres ved hjælp af et såkaldt *rutediagram,* der viser, hvorledes de enkelte led i en beregningsgang hænger sammen. F.eks. kan den netop beskrevne algoritme omskrives til rutediagrammets form på denne måde:


START
a lagres i R_A, -b i R_B og c i R_C
$R_B/2R_A$ lagres i R_D
$R_B^2 - 4 \times R_A \times R_C$ lagres i R_E
$\frac{\sqrt{|R_E|}}{2 \times R_A}$ lagres i R_B
1
$R_E \geqslant 0$?
JA
NEJ
Udlæs R_B
Udlæs R_D
Læg R_B til R_D og udlæs summen
Skift fortegn for R_B
nej
Er begge rødder udlæst?
JA
STOP

Ved første øjekast kan rutediagrammet virke indviklet og uoverskueligt, men det viser sig hurtigt, at opbygningen er klar og logisk. Man starter ved START og følger pilene, der normalt bevæger sig oppe fra og ned. Bemærk, at boksen mærket # 1 giver mulighed for et valg. Der stilles et spørgsmål, og svaret er bestemmende for, hvilken af de to pile fra rhomben, man skal følge. Hvis svaret på spørgsmålet er JA, følges pilen til højre, er det nej, følges pilen, der peger direkte ned.
Som De formodentlig kan se allerede nu, kan rutediagrammet være et nyttigt værktøj til anskueliggørelse af et programforløb. Specielt når det drejer sig at holde hovedet klart i forbindelse med hop og betingede forgreninger, er rutediagrammerne uovertrufne.

Underprogrammer

Måske blev De forbløffet over matrixprogrammets længde. Det er så langt, at det ikke kan lagres på én gang i HP-11C. Hvad der ikke er umiddelbart klart er, at programmet i virkeligheden er enormt kompakt. Det har man bl.a. kunnet gøre ved at udnytte, at en mængde operationer gentages forskellige steder i beregningen. Selv matrixinverteringen gentages. Derfor har man defineret og brugt denne og adskillige mindre programafsnit som underprogrammer.

Teknisk set er et underprogram ethvert sæt instruktioner, der indledes med en etiket f LBL »etiket« og afsluttes med g RTN . LBL »etiket« gør det muligt at komme ind i underprogrammet fra et andet sted i programmet, og RTN gør det muligt at komme tilbage til det punkt, hvor underprogrammet blev aktiveret.

Et underprogram aktiveres eller *kaldes* ved hjælp af GSB »etiket«. I matrixprogrammets hovedafsnit bliver underprogrammet med navnet LBL 8 hentet otte gange. Hver gang hopper programpilen fra GSB 8 til LBL 8, udfører instruktionerne mellem LBL 8 og den nærmest følgende RTN, hvorefter den returnerer til programlinien umiddelbart efter den GSB 8-instruktion, der startede kaldet af underprogrammet.

Underprogrammernes værdi som pladsbesparende elementer i et omfattende program fremgår tydeligt af programmet. Det er langt mere pladsøkonomisk at bruge GSB 8 otte gange end at lagre instruktionerne mellem f LBL 8 og g RTN otte gange i programmet. Men dertil kommer, at programmet vinder i overskuelighed, når man bruger underprogrammet. Når hovedprogrammet kalder et underprogram, optræder kal-

det som en overordnet ordre, der angiver, *hvad* der skal gøres – først i underprogrammet specificeres, *hvordan* det skal gøres. Det giver en række fordele i form af forøget læsbarhed af programmet og forenklet fejlretning.

[ISG] med [RCL] [(i)]

Når De gennemgår programoplægget for at finde ud, hvilke afsnit der gentages, bør De også se efter registerkald, hvor adresserne kan følge et bestemt mønster f.eks. [STO] 1 ... [STO] 2 ... [STO] 3 osv. I sådanne tilfælde kan man med fordel indlægge [STO] [(i)] og [RCL] [(i)] i underprogrammerne. I matrixprogrammet bliver denne teknik brugt i vid udstrækning.

Rutine A (linierne 079-101) beregner determinanten af den 3×3 matrix, der er lagret i registrene R_5-$R_{.3}$. Determinanten beregnes ved hjælp af:

$$\begin{vmatrix} R_5 & R_6 & R_7 \\ R_8 & R_9 & R_{.0} \\ R_{.1} & R_{.2} & R_{.3} \end{vmatrix} = \begin{aligned} & R_5(R_9 \times R_{.3} - R_{.0} \times R_{.2}) \\ & -R_6(R_8 \times R_{.3} - R_{.0} \times R_{.1}) \\ & +R_7(R_8 \times R_{.2} - R_9 \times R_{.1}). \end{aligned}$$

som kan omskrives til:

$$\begin{aligned} & -(R_{.0} \times R_{.2} \times R_5 + R_8 \times R_{.3} \times R_6 + R_{.1} \times R_9 \times R_7) \\ & \qquad + R_7 \times R_{.2} \times R_8 + R_5 \times R_{.3} \times R_9 + R_6 \times R_{.1} \times R_{.0} \end{aligned}$$

Herved opnås ikke blot et klart [×] [x] [+] -mønster, men også en ubrudt række af registerkald fra R_5 til R_0. Disse mønstre bruges i underprogrammet LBL 9 (linierne 001-009).

I dette underprogram bruges [f] [ISG] til at inkrementere indeksregistret (se side 122). Hvert kald af underprogrammet lægger 1 til tallet i indeksregistret. Derefter fremkaldes det dataregister, hvis adresse er lagret i R_I.

Da decimaldelen af tallet i R_I er nul, og heltalsdelen er større end 0, bliver linien efter [f] [ISG] altid sprunget over. Derfor er der lagt en blind instruktion ind efter [f] [ISG] (det er ganske uden betydning, at [f] [PSE] er valgt).

Nedenstående eksempel viser, hvordan underprogrammet virker:

		Første udførelse	Anden udførelse
[LBL] [A]			
4			
[STO] [I]			
0			
[RCL] [·] 0			
[RCL] [·] 2			
[GSB] 9	[LBL] 9		
[RCL] 8	[ISG]	I = 5	I = 6
[RCL] [·] 3	[PSE]	Springes over	Springes over
[GSB] 9	[RCL] [(i)]	Fremkald R_5	Fremkald R_6
osv.	[×]	$R_{.2} \times R_5$	$R_{.3} \times R_6$
	[×]	$R_{.0} \times R_{.2} \times R_5$	$R_8 \times R_{.3} \times R_6$
	[+]	$0 + R_{.0} \times R_{.2} \times R_5$	Lægges til forrige total.
	[STO] 0	Total lagres i R_0	Ny total lagres i R_0.

Pladsbesparelsen er åbenbar. Fire linier skulle have været udført 6 gange – ialt 24 linier. Underprogrammet har reduceret dette til 10 linier (inklusive 4 [STO] [I] og eksklusive [STO] 0, som skulle have været udført under alle omstændigheder).

Indlæsning af data

En del af problemformuleringen tager sigte på at beslutte, hvordan inddata og mellemresultater skal lagres.

I trekantsprogrammet bruges tre variable i hvert af de fem tilfælde: SSS, VSV, SVV, SVS og SSV. Det er klart, at vi har brug for fem indlæsningsrutiner, som nok er forskellige, men ikke desto mindre må have nogle stærke fællestræk. Da HP-11C har fem brugerdefinérbare taster (A-E), kan der afsættes en tast til hvert af de fem trekantstilfældes variabelindlæsning.

Dataindlæsning ved hjælp af de brugerdefinérbare taster kan foregå på et utal af måder. Man kan f.eks. bruge en metode, som passende kan kaldes »lagring med stop«: En brugerdefinérbar tast bruges til at vælge opgavetype og til lagring af det første tal. Derefter udføres en [R/S], som overgiver kontrollen af tastaturet. Efter indtastning af det næste tal trykkes på [R/S], hvorefter programmet lagrer det netop indtastede tal, stopper med R/S osv., indtil det sidste tal er indtastet. Mere raffinerede versioner af denne metode gør det muligt at lade maskinen starte en slags dialog, hvor man f.eks. kan rette fejlindtastede tal (se LBL A i »Løsning af 3 ligninger med 3 ubekendte«.)

I trekantsprogrammet har vi valgt en anden metode, hvor data lagres i rullestakken, inden der trykkes på den brugerdefinérbare tast, som svarer til opgaven. Programmet lagrer dernæst de indtastede tal ved hjælp af instruktionerne: [STO] R_A [R↓] [STO] R_B [R↓] [STO] R_C. Det er en metode, der kun kan bruges, hvis der er et begrænset antal data, men til gengæld er den hurtig og bekvem og optager ikke megen programplads.

Løkker

Newton-Raphson-programmet bruges til numerisk løsning af ligninger, som det ellers ville være vanskeligt eller umuligt at løse. Metoden kan i det fleste tilfælde finde en *tilnærmet* rod; der er ingen garanti for, at metoden finder alle rødder, men når en rod findes, er den i almindelighed ganske nøjagtigt bestemt.

Metoden bruges ved, at man vælger en værdi x, for hvilken f(x) eksisterer. x forsøges valgt således, at f(x) er nær nul eller i det mindste således, at x ligger nær en rod. Dernæst bruges Newton-Raphson-formlen til at finde et nyt gæt – som i almindelighed vil være bedre end det oprindelige. Dette gøres ved hjælp af programmet, som også fortsætter *iterationen,* dvs. fortsætter med at forbedre de producerede tilnærmelser ved hjælp af formlen. Denne proces fortsætter inden for de afsatte grænser, indtil forskellen på den netop beregnede tilnærmelse og den foregående er mindre end en specificeret grænse.

Der er altså en del af programmet, der skal gentages igen og igen, indtil vi enten når det ønskede resultat eller må opgive at finde en løsning. En sådan programdel kaldes en »løkke«, og i dette program udgøres den af linierne 026-052.

For at forhindre at programmet »kører i uendelig løkke«, skal der angives en grænse for antallet af gennemløb således, at programmet stopper, hvis der ikke findes en tilstrækkeligt god tilnærmelse efter et vist antal iterationer. Dette kan gøres på forskellige måder. Man kunne inkrementere et register efter hvert gennemløb, sammenligne indholdet med en maksimumværdi og hoppe ud af løkken eller fortsætte i den. I det foreliggende program lagres maksimumantallet i et register, som derefter dekrementeres efter hvert gennemløb (linie 051). Herved bliver det lettere at ændre det største antal iterationer. Bemærk, at dette ikke har nogen som helst betydning for nøjagtigheden og heller ikke kan bruges til at sikre fundet af en god tilnærmelse. Funktionen kan være så sær og startgættet så dårligt tilpasset, at man efter 100 iterationer stadig er langt fra en rimeligt god tilnærmelse.

De bør altså være et stopkriterium, der ikke afhænger af antallet af gennemløb. Som allerede nævnt kan man sammenligne de to sidst beregnede x-værdier og stoppe kørslen, hvis forskellen ikke er signifikant.

Som kriterium bruges en teststørrelse, Δx, hvis størrelse bestemmes af brugeren i forbindelse med initialiseringen af programmet.

Løkken opbygges da således:

1. Δx og startgættet lagres.
2. Første tilnærmelse beregnes og lagres.
3. Næste tilnærmelse beregnes på basis af den foregående tilnærmelse og lagres.
4. Den numeriske værdi af differensen mellem den aktuelle tilnærmelse og den foregående beregnes.
5. Δx og differensen sammenlignes.
6. Hvis differensen er større end teststørrelsen, fortsættes løkken fra trin 3.
7. Hvis differensen er mindre end eller ligemed teststørrelsen, hoppes ud af løkken.

Bemærk, at heller ikke dette kriterium alene sikrer mod kørsel i uendelig løkke; hvis der f.eks. ligger mange løsninger lige i nærheden af hinanden,

vil det være meget svært at finde en løsning. Derfor bør man kombinere kriterierne, som det er gjort i programmet.

Newton-Raphson-iteration er bl.a. baseret på en beregning af den første afledte for de forskellige x-værdier. Hvis $|F'(x_i)|$ bliver meget stor, kan det ske, at forskellen mellem x_i og x_{i+1} bliver mindre end teststørrelsen – uden at man af den grund har fundet en rod.

For at forhindre programmet i at stoppe under sådanne omstændigheder er der indlagt en ekstra test, hvor $|F(x)|$ sammenlignes med Δx.

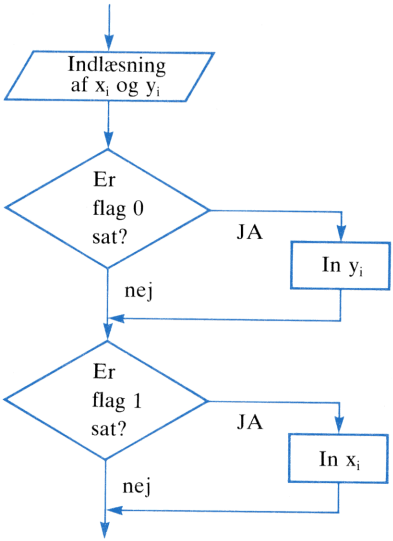
Flag

Nogle programmer bruges til at beslægtede problemer, hvor forskellene ikke er så forfærdelig store, men alligevel så betydende, at man ikke umiddelbart kan anvende underprogrammer. I sådanne tilfælde kan man ofte bruge flag til at styre behandlingen af de forskellige tilfælde. Flag bruges til at opbevare information om en hændelse eller et valg; de er med andre ord statusindikatorer. Hvis programmet f.eks. vil *sætte* flag 1, hvis der foretages en bestemt initialisering, kan man senere ved at *teste* flag 1 afgøre, om denne initialisering har fundet sted.

I kurvetilpasningsprogrammet skal der vælges mellem tre processer, der har meget tilfælles. Flagene bruges til at indikere, hvilken af de tre processer der skal følges. Det gøres på følgende måde (hvor skemaet fra side 155 er udvidet med to søjler):

Regressionstype	**A**	**X_i**	**Y_i**	**Kode**	**Flag 0**	**Flag 1**
Eksponentiel	ln a	x_i	ln y_i	1	Sat	Slettet
Logaritmisk	a	ln x_i	y_i	2	Slettet	Sat
Potens	ln a	ln x_i	ln y_i	3	Sat	Sat

De to flag sættes i starten af programmet og bruges undervejs til at afgøre, hvordan data skal behandles. Processen kan anskueliggøres af følgende rutediagram:



Resultatet (A) behandles på tilsvarende måde. Vi har altså fået løst en opgave, hvor kernen (summationerne) er ens, men hvor programmet skal indledes og afsluttes afhængigt af, hvilken af de tre muligheder der er valgt.

Som det ses af skemaet, kan man vælge mellem *fire* tilstande ved hjælp af *to* flag.

Tilfældige tal

HP-11C har en indbygget generator for pseudotilfældige tal. Når vi siger *pseudo*tilfældige, hænger det sammen med, at ingen datamat kan frembringe rigtig tilfældige tal ved hjælp af en *aritmetisk* proces. Det ligger i maskinens natur, at man altid vil kunne rekonstruere rækken af tal – og derfor også vil kunne forudsige forløbet af en sekvens. HP-11C's generator skal have en udgangstal, hvorefter der produceres et pseudotilfældigt tal mellem 0 og 1 (1 dog undtaget). Som det fremgår af fodnoten i forbindelse med brugerhåndbogens omtale af [f] [RAN#] , har metoden været underkastet spektraltesten, der er brugt til at afgøre, om generatorens resultater fænomenologisk adskiller sig fra rigtig tilfældige tal.

Tilfældige tal bruges først og fremmest inden for den teoretiske og den anvendte statistik. Men også til spil og anden tidsfordriv bruges pseudotilfældige tal til at frembringe praktisk talt uforudsigelige kombinationer. Herved bliver det muligt at give den menneskelige spiller en interessant modspiller, som kan stille egentlige krav. U-bådsspillet er et godt eksempel på et spil, hvor maskinen bruger pseudotilfældige tal til at skabe et dynamisk spil – i dette tilfælde kan der endog vælges mellem forskellige dynamikgrader.

Der skal bruges to pseudotilfældige tal, x- og y-koordinaterne, som skal ligge mellem 0 og 9. Det betyder, at det genererede tal skal ganges med 10. Ved at benytte denne metode kan man skalere til et vilkårligt interval. Hvis De f.eks. skal bruge tilfældige tal i intervallet (34,5;98,36), skal der genereres pseudotilfældige tal mellem 0 og 63,86 (differensen mellem 98,36 og 34,50). Til disse tal lægges 34,5.

Pseudotilfældige tal i intervallet genereres altså således (efter udgangstal [STO] [RAN#]):
[f] [RAN#] (b-a) [×] a [+] for hvert enkelt af tallene.

Brugerdefinérbare taster

De brugerdefinérbare taster [A], [B], [C], [D] og [E] er blandt de mest anvendelige af HP-11C's faciliteter. De er specielt anvendelige til tre typer operationer: 1) Lagring af data; 2) Valg mellem forskellige rutiner i et program; 3) Valg mellem forskellige programmer, der er lagret samtidigt i programhukommelsen.
Finansprogrammet udnytter de to første af disse muligheder.

Lagring af data

Finansprogrammet arbejder med fem forskellige inddata: rentesats *(i)*, antal terminer *(n)*, hovedstol eller nutidsværdi *(PV)*, fast ydelse *(PMT)* og slutværdi eller restgæld *(FV)*. Da der netop er fem brugerdefinérbare taster, kan man tildele hver af dem en parameter. Det betyder, at f.eks. tasten [C] aktiverer delprogrammet [f] [LBL] [C] [STO] 3 [R/S]; hver gang der trykkes på [C], lagres tallet i lyspanelregistret i R_3. Blandt fordelene ved denne metode er, at værdierne kan indtastes i vilkårlig rækkefølge, og at de er lette at rette.

Valg af forskellige rutiner

Et program kan skrives, så det kan beregne mere end én værdi, og man kan benytte de brugerdefinérbare taster til at vælge, hvilken værdi der skal beregnes. I finansprogrammet benytter man de brugerdefinérbare taster, ikke bare til indlæsning af data, men også til beregning af de fem parametre. Skal man f.eks. have beregnet *PMT*, trykker man på [D] og på [R/S], når programmet stopper. Når der trykkes på [D], bliver det udlæste tal lagret i R_4, hvis gamle værdi altså bliver overskrevet. Men det gør ingenting, for der skal jo netop beregnes en ny værdi af *PMT*, og når programmet stopper, vil brugen af [R/S] medføre, at *PMT* beregnes, og at resultatet lagres i R_4.

Tillæg A

Fejlmeddelelser

Hvis De bruger en af nedennævnte taster under den eller de betingelser, der nævnt, vil der fremkomme en fejlmeddelelse på formen ***Error n*** ($0 \leqslant n \leqslant 9$) som tegn på, at en ugyldig eller udefineret operation er forsøgt udført. En fejltilstand og dens tilhørende fejlmeddelelse ophæves ved tryk på en vilkårlig tast.

Error 0: Matematiske funktioner

[÷] når $x = 0$.
[y^x] når $y = 0$, $x \leqslant 0$ eller $y < 0$ og x ikke er et heltal.
[$\sqrt{x}$] når $x < 0$.
[1/x] når $x = 0$.
[LOG] når $x \leqslant 0$.
[LN] når $x \leqslant 0$.
[SIN^{-1}] når $|x| > 1$.
[COS^{-1}] når $|x| > 1$.
[HYP^{-1}] [COS] når $|x| < 1$.
[HYP^{-1}] [TAN] når $|x| > 1$.
[STO] [÷] når $x = 0$.
[△%] når $y = 0$.
[Py,x] når
1. x og/eller y ikke er heltal.
2. $x \leqslant 0$ og/eller $y \leqslant 0$.
3. $x > y$.
4. $x \geqslant 10^{10}$.

[Cy,x] – samme som for [Py,x]

Error 1: Dataregistre

Overløb i dataregister (undtagen efter [Σ+] og/eller [Σ−]). Tallet i registret er numerisk større end $9{,}999999999 \times 10^{99}$.

Error 2: Statistik

[x̄] når $n = 0$.
[s] når $n \leq 1$.
[ŷ,r] når $n \leq 1$.
[L.R.] når $n \leq 1$.

Bemærk: ***Error 2*** udlæses også ved forsøg på division med 0 eller kvadratrodsuddragning af et negativt tal under beregning af følgende formler:

$$s_x = \sqrt{\frac{M}{n(n-1)}} \qquad s_y = \sqrt{\frac{N}{n(n-1)}} \qquad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \qquad B = \frac{M\,\Sigma y - P\,\Sigma x}{n \cdot M}$$

(A og B er de koefficienter, der beregnes med [f] [L.R.]).

$$\hat{y} = \frac{M\,\Sigma y + P\,(n \cdot x - \Sigma x)}{n \cdot M}$$

hvor

$M = n\,\Sigma x^2 - (\Sigma x)^2$
$N = n\,\Sigma y^2 - (\Sigma y)^2$
$P = n\,\Sigma xy - \Sigma x\,\Sigma y$

Error 3: Registeradressering

Det kaldte register er:
1. Konverteret til programlinier.
2. Uden for de definerede grænser (0 til .9).

Error 4: Programadressering

Den kaldte *programlinie* er:
1. Optaget som dataregister.
2. Uden for de definerede grænser (0 til 203).

Den kaldte *etiket* eksisterer ikke.

Error 5: Underprogrammer

Overløb i returadressestakken for underprogrammer: der er allerede kaldt underprogrammer i fire niveauer.

Error 6: Flag

Flagadressen er ikke 0 eller 1.

Error 7 og Error 8

Bruges ikke.

Error 9: Service

Fejl under selvtest. Se Tillæg B, »Batteri, service og udbedring af mangler«.

Pr Error: Kontinuerlig hukommelse

Den kontinuerlige hukommelse er slettet på grund af strømsvigt.

Tillæg B

Batteri, service og udbedring af mangler

HP-11C drives af tre batterier. Maskinen leveres med 3 alkaliske knapceller, men også sølvoxidbatterier kan bruges.

Et sæt batterier bestående af 3 friske alkaliske knapceller holder til mindst 80 timers uafbrudt brug. Hvis maskinen ikke bruges, vil batterierne alligevel ikke holde uendeligt; de udsættes for en vis selvafladning. Den kontinuerlige hukommelse påvirker ikke batterierne i nogen betydende grad; det betyder, at den kontinuerlige hukommelses indhold bevares indtil batteriernes klemspænding pludselig falder drastisk – det sker tidligst efter 1½ år.

Et sæt friske sølvoxidbatterier holder til mindst 180 brugstimer. Sølvoxidbatterierne selvaflades lidt langsommere end de alkaliske celler, så man kan regne med, at den kontinuerlige hukommelse ihvertfald bevares i 2 år, hvis maskinen ikke bruges.

Den faktiske batterilevetid vil ligge et eller andet sted imellem disse yderpunkter og afhænger ganske af brugen: om maskinen bruges i længere perioder ad gangen, om den bruges manuelt eller til programkørsel. Levetiden afhænger også af brugstemperaturen: batterierne aflades meget hurtigt, hvis de bruges ved lave temperaturer, og selvafladningen foregår hurtigst ved høje temperaturer. De almindelige knapceller er beregnet til brug ved stuetemperatur.

Bemærk, at batterierne *ikke* er genopladelige.

Advarsel

Batterierne kan eksplodere under voldsom trykudvikling, hvis De forsøger at genoplade batterierne, opbevarer dem nær en hed varmekilde eller kaster dem i åben ild.

De kan bruge batterier af eller tilsvarende mærker i HP-11C:

Alkaliske

IEC LR44 – UCAR A76
Ray-O-Vac RW 82
BEREC LR44
Varta 4276 – Toshiba LR44

Sølvoxid

IEC SR44 – UCAR 357
Ray-O-Vac RS 76 eller H-RW42
BEREC BSR44H eller BSR44L
Varta 541 – Toshiba G13
Mallory 10L14

Batteriindikator

En stjerne (*), der blinker i lyspanelets nederste venstre hjørne, er tegn på, at batterierne skal udskiftes.

HP-11C kan dog bruges endnu nogen tid efter at batteriindikatoren bliver tændt. De opgivne brugstider gælder for kontinuerlig programkørsel; ved blandet drift (dvs. med tænd og sluk), bliver brugstiden forlænget.

Ved brug af alkaliske batterier gælder:

- HP-11C kan bruges i mindst 2½ time efter at stjernen viser sig første gang.
- Hvis man slukker for maskinen, når stjernen begynder at blinke, kan den kontinuerlige hukommelses indhold bevares mindst 1 uge endnu.

Ved brug af sølvoxidbatterier gælder:

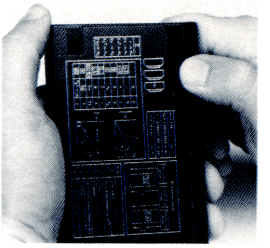
- HP-11C kan bruges i mindst 15 minutter efter at stjernen viser sig første gang.
- Hvis man slukker for maskinen, når stjernen begynder at blinke, kan den kontinuerlige hukommelses indhold bevares mindst 3 dage endnu.

Udskiftning af batterier

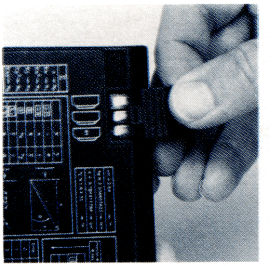
Når De tager batterierne ud af HP-11C, vil den kontinuerlige hukommelses indhold blive bevaret så længe, at De skulle have rigelig tid til at sætte nye batterier på plads.
Hvis der går mere end ½ minut, risikerer De dog at slette den kontinuerlige hukommelse.

Batterierne udskiftes på følgende måde:

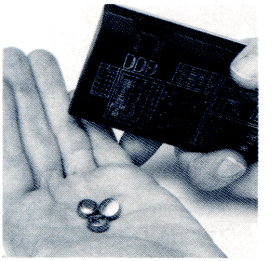
1. Sørg for at maskinen er slukket.
2. Hold maskinen som vist på figuren, og tryk fremad på batteridækslet, indtil det glider løs.
3. Tag fat i batteridækslets fri ende, og løft dækslet af.



Bemærk: Pas på, at De ikke kommer til at trykke på nogen af HP-11C's taster under udførelsen af resten af batteriskiftet. Hvis De kommer til at aktivere en tast, risikerer De, at den kontinuerlige hukommelse bliver slettet, og De risikerer at HP-11C ikke reagerer på tastaturoperationer.



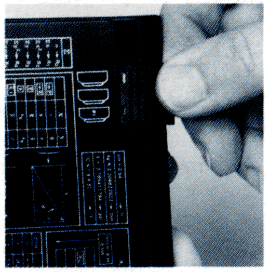
4. Vend regnemaskinen om, og slå den forsigtigt mod håndfladen, så batterierne falder ud.



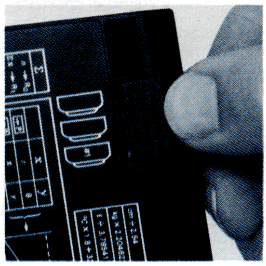
Bemærk: Pas på, at De ikke sætter batterierne omvendt på plads. Hvis det sker, risikerer De at slette den kontinuerlige hukommelse.

5. Sæt *tre* nye batterier på plads i batterirummet, og sørg endelig for, at batterierne bliver placeret som vist på tegningen under batterirummet. + polerne (de flade sider) skal vende mod nærmeste gummifod.

6. Skub batteridækslets lange tap ind i slidsen under kabinettets bagside.



7. Lad dækslet falde til og pres det ind mod kabinettet, indtil det sidder helt fast.
8. Tryk på [ON], så der tændes for maskinen. Hvis den kontinuerlige hukommelse af en eller anden grund er blevet slettet, vises ***Pr Error*** i lyspanelet.



Selvtest

Hvis det virker, som om maskinen ikke fungerer rigtigt eller ikke vil tænde, kan De forsøge følgende:

1. Hvis maskinen ikke reagerer på indtastninger, skal De først tage batterierne ud og sætte dem på plads igen. Sørg for, at batterierne sidder rigtigt i batterirummet: med polerne i samme retning som vist på stregtegningen ved siden af batterirummet. Hvis det ikke hjælper, bør De forsøge igen efter at have udskiftet batterierne med nye batterier.

2. Sluk for HP-11C. Tryk på [ON] og hold tasten nedtrykket medens De trykker på [×].

3. Hvis HP-11C ikke reagerer på indtastninger, kan De forsøge at ***kortslutte*** terminalerne et kort øjeblik. Løft plasticflapperne op, og hold en clips eller lignende mod metalkontakterne i batterirummet. Det er nok med en ganske kort berøring. Kortslutningen nulstiller den kontinuerlige hukommelse.

4. Slip først [ON] og dernæst [×]. Herved startes en automatisk selvtest af alle HP-11C's elektroniske kredse. Hvis alt er i orden, vil den efter ca. 15 sekunders forløb (hvor ***running*** vises i lyspanelet) stoppe selvtesten og udlæse -8, 8, 8, 8, 8, 8, 8, 8, 8, 8, samt alle lyspanelets statusindikatorer (undtagen *) – inklusive nogle, der ikke normalt vises, men findes tilgængelige i det lyspanel, der bliver brugt. Hvis der udlæses ***Error 9,*** hvis lyspanelet er tomt eller der kommer et andet resultat, skal maskinen sendes til reparation hos Hewlett-Packard.*).

Bemærk: Også [+] og [÷] udfører interne checkrutiner, når de bruges på ovennævnte måde sammen med [ON].**) Disse tests bruges til at kontrollere kredsløbene under fremstillingsprocessen og ved service.

Hvis De har haft mistanke til maskinen, men trin 4 i selvtesten giver den rette udlæsning, er der stor sandsynlighed for, at De har fejlbetjent ma-

---

*) Hvis [ON] / [×] eller [ON] / [+] giver ***Error 9,*** kan De fortsætte med at bruge maskinen efter at have slettet den kontinuerlige hukommelse.

**) [ON] / [+] starter en test, der svarer til den ovennævnte, men som blot fortsætter, indtil De stopper den ved at trykke på en eller anden tast (der kan gå op til 15 sekunder, før testen afbrydes efter tryk på en tast).

[ON] / [÷] bruges til en test af lyspanel og tastatur. Når [ON] slippes, tændes nogle enkelte af lyspanelets segmenter.

Testen udføres ved, at man trykker på alle tasterne, en efter en, i en bestemt rækkefølge: man starter med tasten yderst til venstre i øverste række, fortsætter med den næste tast i samme række. Således fortsætter man, indtil alle taster er brugt. Hvis maskinen virker, og *man har trykket på tasterne i den rigtige rækkefølge,* vil der stå 11 i lyspanelet, når testen er forbi. ([ENTER] skal bruges både under test af tredie række og under fjerde række). Hvis maskinen ikke virker efter hensigten, eller *hvis tasterne ikke er trykket i den rigtige rækkefølge* (og det er langt mere sandsynligt), vil ***Error 9*** blive vist i lyspanelet.

Hvis De får udlæst ***Error 9*** efter denne test, bør De gennemgå den mindst én gang til: det sker meget let, at man glemmer en tast, trykker to gange på en tast eller på anden måde gør noget forkert.

skinen. Vi foreslår, at De genlæser det relevante afsnit i brugsanvisningen. Hvis De stadig får resultater, der ikke kan være rigtige, beder vi Dem sende maskinen til eftersyn hos Hewlett-Packard.

Udbedring af mangler

Såfremt der konstateres materiale- og/eller fabrikationsmangler, forpligter Hewlett-Packard sig til uden beregning at udbedre sådanne mangler på betingelse af, at der rettidigt er reklameret herover inden for en periode på tolv måneder fra købsdatoen, og at lommeregneren returneres med forsendelsesomkostningerne betalt til et af Hewlett-Packard udpeget servicecenter, sammen med en kopi af købskvitteringen som dokumentation for købsdatoen.
Transportskader og forsendelsesomkostninger dækkes ikke. Foranstående forpligtelser gælder ikke, hvis lommeregneren er beskadiget ved uheld, misbrug eller på grund af service eller ændringer foretaget af andre end et autoriseret Hewlett-Packard servicecenter. Eventuelle mangler på forbrugsartikler dækkes ikke af foranstående forpligtelser. *Hewlett-Packard er ikke ansvarlig for følgevirkninger.*

Efter udbedringsperiodens udløb

Efter udbedringsperioden på ét år, repareres lommeregneren til lavest mulige pris. Der er tre måneders udbedringsperiode på alle reparationer, der udføres efter udbedringsperiodens udløb af et autoriseret Hewlett-Packard servicecenter.

Overførsel af ovennævnte forpligtelse

Hvis De sælger Deres lommeregner eller forærer den væk, kan ovennævnte forpligtelser overdrages og gælder stadig for den nye ejer, indtil den oprindelige ét-årige udbedringsperiode er udløbet. Det er ikke nødvendigt at give Hewlett-Packard meddelelse om overdragelsen. Dokumentation for købet (købskvittering) bør dog videregives til den nye ejer.

Specifikationer

Hewlett-Packards produkter sælges på basis af de specifikationer, der er gældende på salgstidspunktet. Hewlett-Packard har ingen forpligtelse til at ændre eller modernisere produkter, der er solgt og betalt.

Forsendelsesvejledning

Lommeregneren returneres sammen med det udfyldte servicekort i salgskartonen (eller anden emballage, der yder tilsvarende beskyttelse) for at undgå skader under transporten. Transportskader dækkes ikke af Hewlett-Packard, som derfor anbefaler, at alle forsendelser forsikres særskilt. Hvis lommeregneren skal repareres vederlagsfrit i henhold til ovennævnte forpligtelse, skal De vedlægge dokumentation for købsdatoen.

Lommeregneren skal returneres til nærmeste af Hewlett-Packard udpegede servicecenter. Kontakt Deres HP-forhandler for nærmere oplysninger.

Yderligere oplysninger

Oplysninger om lommeregnerens kredsløb og design betragtes som Hewlett-Packards ejendom. Heller ikke servicehåndbøger kan stilles til Deres rådighed. Hvis der skulle opstå andre problemer, bedes De kontakte Deres HP-forhandler eller det nærmeste Hewlett-Packard kontor.

Temperaturspecifikationer

- Brugstemperatur: 0° til 55°C
- Lagertemperatur: ÷40° til 65°C

International serviceorganisation

Ikke alle Hewlett-Packard servicecentre kan yde service på alle typer lommeregnere fra Hewlett-Packard. De kan dog være sikker på at kunne få repareret Deres lommeregner i det land, hvor den er købt. Hvis De opholder Dem uden for indkøbslandet, kan De kontakte det lokale Hewlett-Packard kontor, hvor det vil kunne oplyses, om lommeregneren kan repareres på stedet.

Alle nedennævnte Hewlett-Packard servicecentre i Europa kan udføre service på alle lommeregnere fra Hewlett-Packard.

Også for service i et andet land end indkøbslandet gælder ovennævnte forpligtelser. *Det bemærkes specielt, at De skal kunne dokumentere købsdatoen for at kunne opnå vederlagsfri udbedring af mangler.*

Servicecentre i Europa

BELGIEN
HEWLETT-PACKARD BELGIUM SA/NV
Boulevard de la Woluwe 100
Woluwelaan
B-1200 BRUXELLES
Telefon: (2) 762 32 00

DANMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 BIRKERØD
Telefon: (02) 81 66 40

FINLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Telefon (90) 455 02 11

FRANKRIG
HEWLETT-PACKARD FRANCE
S.A.V. Calculateurs de Poche
Les Ulys
Boite Postale No 6
F 91401 ORSAY CEDEX
Telefon: (1) 907 78 25

HOLLAND
HEWLETT-PACKARD NEDERLAND B.V
Van Heuven Goedhartlaan 121
1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 667
Telefon: (020) 472021

ITALIEN
HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. de Vittorio, 9
I 20063 CERNUSCO SUL NAVIGLIO (Milano)
Telefon: (2) 90 36 91

NORGE
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Østerndalen 18
N 1345 ØSTERÅS (Oslo)
Telefon: (2) 17 11 80

SCHWEIZ
HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH 8967 WIDEN
Telefon (057) 50111

SPANIEN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
MADRID 16
Telefon: (1) 458 2600

SVERIGE
HEWLETT-PACKARD SVERIGE AB
Enighetsvägen 3
Box 205 02
S 161 BROMMA 20 (Stockholm)
Telefon: (8) 730 05 50

STORBRITANNIEN
HEWLETT-PACKARD Ltd
308/314 Kings Road
GB READING, Berks.
Telefon: (734) 61022

VESTTYSKLAND
HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Berner Strasse 117
Postfach 560 140
D 6000 FRANKFURT 56
Telefon: (611) 50041

ØSTLANDENE
Se adressen under ØSTRIG

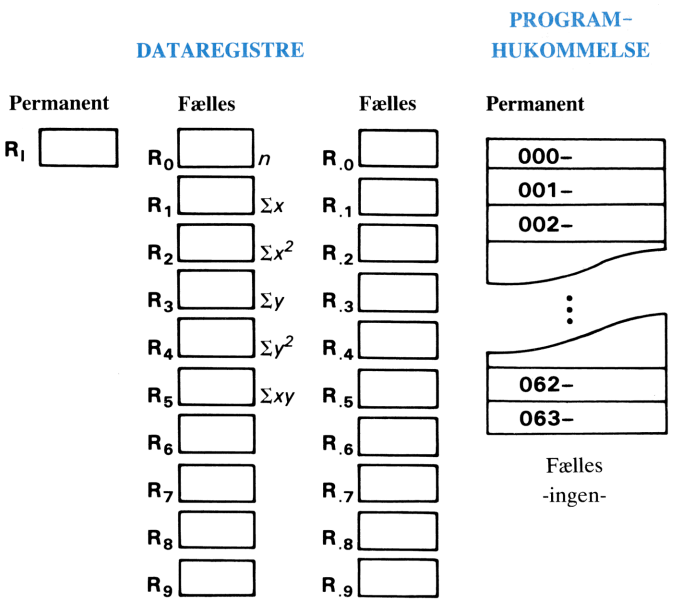
ØSTRIG
HEWLETT-PACKARD GmbH
Kleinrechner-Service
Wagramerstr.-Lieblgasse
A 1220 WIEN
Telefon: (222) 23 65 11

Tillæg C

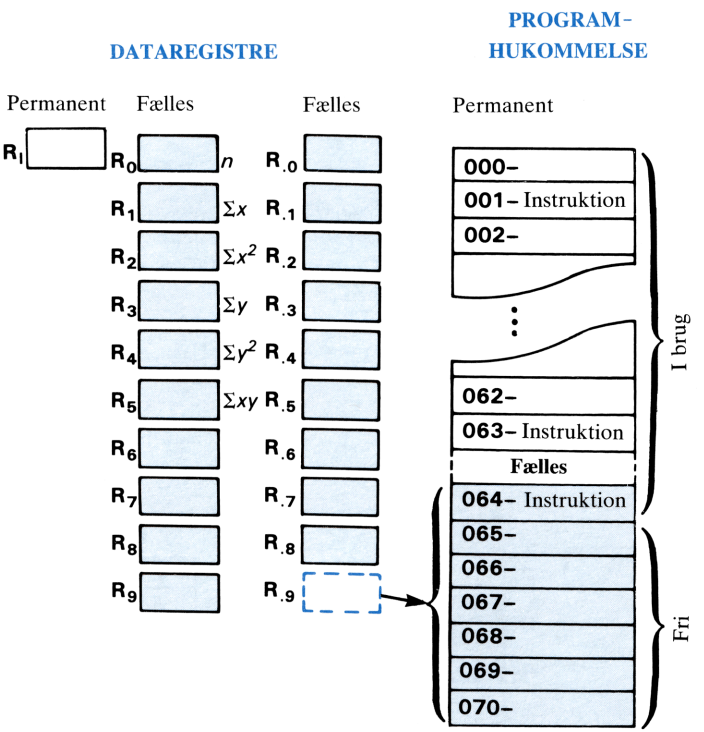
Automatisk splitning

Konvertering af dataregistre til programlinier

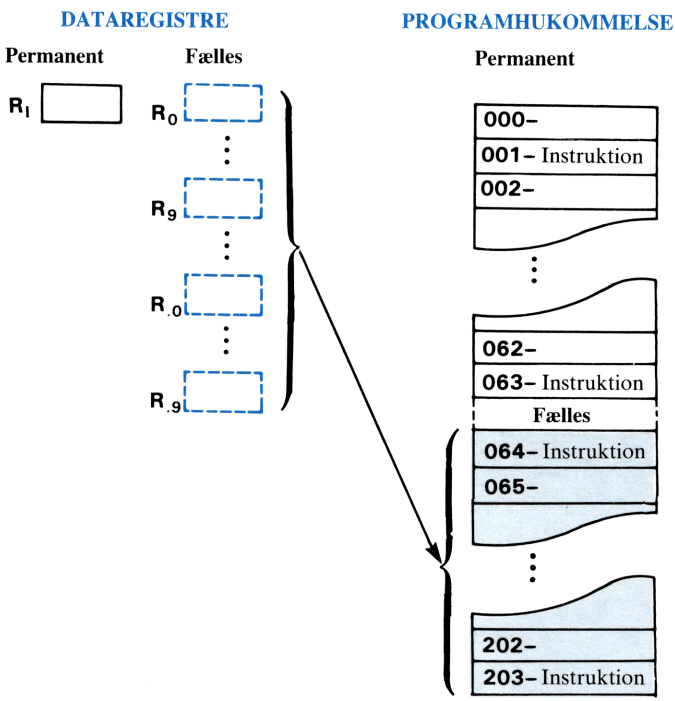
HP-11C's automatiske splitning giver Dem forøget anvendelighed af hukommelsen, idet maskinen selv sørger for at stille det nødvendige antal programlinier til rådighed, samtidig med at der beholdes så mange dataregistre som muligt. Der startes med 63 programlinier og 20 dataregistre (samt indeksregistret, der er beskrevet i kapitel 2, afsnit 5). Når der er lagret fra 0 til 63 instruktioner i programhukommelsen, ser splitningen således ud:



Når den 64. programlinie indtastes, konverteres dataregister $R_{.9}$ til 7 programlinier. Derefter ser splitningen således ud:



Når der er lagret 203 programlinier, opnås følgende splitning:



Bemærk, at der nu kun er det ikke-konvertible indeksregister tilbage af de oprindelige 21 dataregistre (R_0-$R_{.9}$ og R_1). Alle de konvertible dataregistre er nu konverteret til programhukommelse – dette er en højst usædvanlig splitning, som de næppe nogen sinde vil bruge; normalt skal der også bruges nogle dataregistre – men muligheden for at konvertere hele hukommelsen foreligger altså. I kapitel 2, afsnit 1 kan De se en oversigt over samhørende dataregistre og programlinier.

Som De kan se, er splitningen fuldautomatisk; så snart den tilgængelige programplads er brugt, konverteres et dataregister til 7 programlinier. F.eks. vil lagring af programlinie 71 medføre, at R_8 konverteres til programlinierne 71-78.

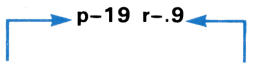
Bemærk: HP-11C konverterer dataregistrene oppefra og nedefter således, at $R_{.9}$ konverteres først, dernæst $R_{.8}$ og til allersidst R_0. Af denne grund bør De vælge dataregistre i den modsatte rækkefølge: fra R_0 og opefter, så De ikke risikerer at få konverteret dataregistre, der skulle være brugt til lagring af data. Det er af største betydning at holde sig dette for øje: Er et dataregister først konverteret til programlinier, mistes den information, der var lagret i registret.

Konvertering af programlinier til dataregistre

Ved brug af [f] CLEAR [PRGM] i programmeringsstatus bliver alle fælles programlinier (linier 064-203) konverteret til dataregistre. Dette vil ofte være for drastisk en handling; med [←] kan enkelte programlinier slettes, hvorved der foregår en løbende konvertering (se kapitel 2, afsnit 6).

Brug af [MEM]

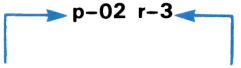
[g] [MEM] beskriver den aktuelle splitning og kan bruges i såvel normal- som programmeringsstatus. Hvis der f.eks. er brugt 44 programlinier, vil [g] [MEM] give følgende i lyspanelet:



Antal instruktioner, der kan indtastes før næste konvertering af et dataregister til syv linier.

Dataregistret, der konverteres næste gang.

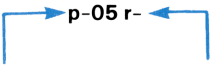
Hvis [g] [MEM] bruges med 173 programlinier, vises:



Antal instruktioner, der kan indtastes før næste konvertering af et dataregister til syv linier.

Dataregistret, der konverteres næste gang.

Bruges [g] [MEM] med 198 programlinier, vises:



Antal instruktioner, der kan indtastes før programhukommelsen er helt fuld.

Der kan ikke konverteres flere dataregistre.

Bemærk: De statistiske funktioner benytter dataregistrene R_0-R_5 som sumregistre. det betyder, at programmer, hvor der bruges statistikfunktioner, højst kan være på 161 programlinier. Forsøg på at udføre statistik med programmer på mere end 161 linier resulterer i fejlmeddelelsen ***Error 3.***

Tillæg D

Stakløft og SIDSTE X

HP-11C er konstrueret, så den fungerer naturligt og letforståeligt. Som de allerede har set under gennemgangen af denne håndbog, behøver De sjældent tænke på, hvordan den automatiske rullestak virker – De skal blot udføre beregningerne på samme måde, som hvis De brugte papir og blyant: én operation ad gangen.

Der er dog nogle tilfælde, særlig under programmering af lommeregneren, hvor det er nødvendigt at have fuldt overblik over rullestakkens indhold og adfærd. Til det formål skulle følgende kunne være Dem til hjælp.

Afslutning af talindtastning

De fleste operationer vil, uanset om de udføres fra tastaturet eller under programkontrol, afslutte en talindtastning. Det betyder i praksis, at maskinen vil opfatte det første ciffer efter alt andet end en talindtastning som første ciffer i et nyt tal. [CHS], [·], [EEX] og [←] afslutter ikke en talindtastning.

Stakløft

HP-11C's operationer kan opdeles i tre grupper alt efter deres indvirken på stakløften: de kan *aktivere* stakløften, *blokere* den eller være *neutrale*.

Operationer, der blokerer stakløften

Når stakløften er blokeret, vil det næste tal *overskrive* indholdet af X-registret. Kun fire operationer blokerer stakløften (se kapitel 1, afsnit 2):

[ENTER] [Σ+] [g] [Σ−] [g] [CLx]

Operationer, der aktiverer stakløften

Når stakløften er aktiveret, vil indtastning af et tal bevirke, at det gamle indhold af X-registret skubbes op i Y-registret (hvis indhold skubbes op

i Z, som skubbes op i T – hvis gamle indhold går tabt). Langt de fleste operationer aktiverer stakløften – det gælder f.eks. alle de matematiske standardfunktioner. Bemærk, at skift fra programmeringsstatus til normalstatus aktiverer stakløften. Når der tændes for maskinen, aktiveres stakløften.

T →				
Z →				
Y →		4.0000	4.0000	53.1301
X →	4	4.0000	3	5.0000
Indtastning	4	ENTER Stakløft blokeres	3 Ingen stakløft	g →P Stakløft aktiveres

T →				
Z →				
Y →	53.1301	53.1301	53.1301	53.1301
X →	0.0000	7	1.0000	9
Indtastning	g CLx Stakløft blokeres	7 Ingen stakløft	Σ+ Stakløft blokeres	9 Ingen stakløft

Neutrale operationer

De neutrale operationer efterlader stakløften i samme stand, som før de blev udført og er altså virkelig neutrale i deres forhold til stakløften. Det drejer sig typisk om funktioner, der ændrer maskinens status men iøvrigt ikke udfører nogen funktion på tal i rullestakken. Hvis man f.eks. har trykket ENTER og dernæst ændrer fastkomma med f FIX, vil stakløften stadig være blokeret. Også de taster, der bruges til talindtastning, er i sagens natur neutrale – dog vil CHS aktivere stakløften, hvis den ikke bruges i umiddelbar forlængelse af en talindtastning.

Følgende operationer er neutrale:

FIX
SCI
ENG
DEG
RAD
GRD
MEM

GTO . **nnn**
BST
SST (I normalstatus, kan SST udføre en funktion, der aktiverer stakløften).
USER

CLEAR PREFIX
CLEAR REG
CLEAR Σ
CHS
R/S
PSE
P/R

SIDSTE X

Følgende operationer lagrer sidste x-værdi i SIDSTE X:

−
+
×
÷
→H.MS
→H
ABS
→R
→D

Σ+
Σ−
%
Δ%
ŷ,r
x!
FRAC
INT
LN

e^x
LOG
10^x
SIN
SIN^{-1}
COS
COS^{-1}
TAN
TAN^{-1}

$\sqrt{x}$
x^2
$1/x$
y^x
→R
→P
$P_{y,x}$
$C_{y,x}$

RND
HYP SIN
HYP COS
HYP TAN
HYP^{-1} SIN
HYP^{-1} COS
HYP^{-1} TAN

Programteknik – indeks

Funktion	Side
Betingelser	
Trekantsberegninger	148
Kurvetilpasning	156
Newton-Raphson-iteration	162
U-bådsjagt	165
Finansregning	178
Numerisk integration	194
[DSE]	
3 ligninger med 3 ubekendte	142
Newton-Raphson-iteration	162
Flag	
Matrixoperationer	134
Kurvetilpasning	156
U-bådsjagt	165
Finansregning	178
χ^2-test	191
[GTO] [I]	
Kurvetilpasning	156
Indlæsningsrutiner	
3 ligninger med 3 ubekendte	142
Trekantsberegninger	148
Kurvetilpasning	156
Newton-Raphson-iteration	162
U-bådsjagt	165
Finansregning	178
t-test	186
Numerisk integration	194

Side

[ISG]
Matrixoperationer 134
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148

Løkker
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148
Kurvetilpasning 156
Newton-Raphson-iteration 162
Finansregning 178
t-test 186
χ^2-test 191
Numerisk integration 194

[L.R.] og beslægtede funktioner
Kurvetilpasning 156
t-test 186
χ^2-test 191

Udlæsningsrutiner
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148
Kurvetilpasning 156
t-test 186

[RAN#]
U-bådsjagt 165

[RCL] [(i)]
Matrixoperationer 134
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148

[STO] [(i)]
3 ligninger med 3 ubekendte 142

[STO] [+] nn
U-bådsjagt 165
χ^2-test 191
Numerisk integration 194

Side

STO × nn
Newton-Raphson-iteration 162

Underprogrammer
Matrixoperationer 134
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148
Newton-Raphson-iteration 162
U-bådsjagt 165
Finansregning 178
Numerisk integration 194

Brugerdefinérbare taster
Matrixoperationer 134
3 ligninger med 3 ubekendte 142
Trekantsberegninger 148
Kurvetilpasning 156
Newton-Raphson-iteration 162
U-bådsjagt 165
Finansregning 178
t-test 186
χ^2-test 191
Numerisk integration 194

$x \gtrless (i)$
3 ligninger med 3 ubekendte 142

Oversigt over funktionstaster

[ON] – Tænder og slukker for lyspanelet. (Se side 16).

Omregninger

[→R] – *(to Rectangular)* – Omregner polære *til* rektangulære koordinater, hvor *(r,θ)* lagret i X- og Y-registrene omregnes til *(x,y)*. (Se side 49).

[→P] – *(to Polar)* – Omregner rektangulære *til* polære koordinater, hvor *(x,y)* lagret i X- og Y-registrene omregnes til *(r,θ)*. (Se side 49).

[→H.MS] – *(to Hours. Minutes Seconds)* – Omregner decimaltimer (eller grader) til timer (grader), minutter og sekunder. (Se side 45).

[→H] – *(to Hours)* – Omregner timer (grader)/minutter/sekunder til decimaltimer (grader). (Se side 45).

[→RAD] – *(to Radians)* – Omregner grader til radian. (Se side 45).

[→DEG] – *(to Degrees)* – Omregner radian til grader. (Se side 45).

Talindtastning

[ENTER] – *(Enter)* – Overfører en kopi af tallet i X-registret til Y-registret, hvis indhold overføres til Z osv. Bruges til at separere tal, der indtastes umiddelbart efter hinanden. (Se side 26).

[CHS] – *(CHange Sign)* – Skifter fortegn på mantissen eller eksponenten i X-registret. (Se side 17).

[EEX] – *(Enter EXponent)* – Skifter fra mantisse- til eksponentindtastning; de følgende cifre betragtes som eksponentcifre. (Se side 66).

[0] til [9] – Taltaster til indtastning af al numerisk information.

[.] – Decimalpunkt (bruges som decimalkomma).

Udlæsningsformater

[FIX] – *(FIXed point)* – Bruges til at vælge fastkomma. (Se side 64).

[SCI] – *(SCIentific notation)* – Bruges til at vælge flydende tal. (Se side 64).

[ENG] – *(ENGineering notation)* – Bruges til at vælge metrisk notation. (Se side 65).

[f] CLEAR [PREFIX] – Udlæser mantissens 10 betydende cifre, uanset hvilket udlæsningsformat der er valgt. Når [PREFIX] slippes, skiftes tilbage til det oprindelige format. Bruges også til at annullere en fejlindtastet forvalgstast. (Se side 68).

Hyperbolske funktioner

[HYP] [SIN], [HYP] [COS] og [HYP] [TAN] – *(HYPerbolic SINe, COSine and TANgens)* – Beregner hyperbolsk sinus, cosinus og tangens af tallet i X-registret. (Se side 46).

[HYP^{-1}] [SIN], [HYP^{-1}] [COS] og [HYP^{-1}] [TAN] – *(inverse HYPerbolic SINe, COSine and TANgens)* – Beregner invers hyperbolsk sinus, cosinus og tangens af tallet i X-registret. (Se side 46).

Indeksregistret

[I] – (Index) – Bruges sammen med [STO] og [RCL] til at adressere indeksregistret, som dels kan bruges som almindeligt dataregister, dels til lagring af det kontroltal, der bruges ved løkker og indirekte operationer. (Se side 121).

[$x \gtrless$ I] – Ombytter tallet i X-registret med tallet i indeksregistret. (Se side 121).

[(i)] – *(indexed)* – Bruges sammen med [STO] og [RCL] til at udføre indirekte lagring, fremkald og registeraritmetik. (Se side 124).

[$x \gtrless$ (i)] – Ombytter tallet i X-registret med tallet i det dataregister, hvis adresse er lagret i R_I. (Se side 124).

[DSE] – *Decrement, Skip if Equal or less than)* – Trækker det specificerede dekrement fra tællerværdien i R_I. Overspringer den efterfølgende programlinie, hvis den nye tællerværdi er lig med eller mindre end den specificerede testværdi. (Se side 122).

[ISG] – *(Increment, Skip if Greater)* – Lægger det specificerede inkrement til tællerværdien i R_I. Overspringer den efterfølgende programlinie, hvis den nye tællerværdi er større den den specificerede testværdi. (Se side 122).

Logaritme- og eksponentialfunktioner

[LN] – Beregner den naturlige logaritme af tallet i X-registret. (Se side 46).

[e^x] – Beregner eksponentialfunktionen af tallet i

X-registret. (Se side 46).

[LOG] – Beregner 10-tals-logaritmen af tallet i X-registret. (Se side 46).

[10^x] – Opløfter 10 til den x'te potens, hvor x er tallet i X-registret. (Se side 46).

[y^x] – Opløfter tallet i Y-registret til den x'te potens, hvor x er tallet i X-registret. (Se side 47).

Aritmetik

[−] [+] [×] [÷] – Aritmetiske operatorer. (Se side 21).

[$\sqrt{x}$] – Uddrager kvadratroden af tallet i X-registret. (Se side 43).

[x^2] – Kvadrerer tallet i X-registret. (Se side 43).

[$x!$] – Beregner fakultet x, x!, eller gammafunktion $\Gamma(1+x)$ af tallet i x-registret. (Se side 42).

[1/x] – Beregner den reciprokke værdi af tallet i X-registret. (Se side 42).

[π] – Lægger en kopi af π (3,141592654) i X-registret. (Se side 41).

Ændring af udlæst tal

[RND] – *(RouND)* – Afrunder tallet i *lyspanelregistret* (X), så det svarer til tallet, der udlæses i *lyspanelet.* (Se side 41).

[ABS] – *(ABSolute value)* – Beregner den numeriske værdi af tallet i X-registret. (Se side 41).

[INT] – *(INTeger)* – Beregner heltalsdelen af tallet i X-registret (dvs. bortkaster decimaldelen). (Se side 41).

[FRAC] – *(FRACtion)* – Beregner decimaldelen af tallet i X-registret (dvs. bortkaster heltalsdelen). (Se side 41).

Procentregning

[%] – Beregner x% af y, hvor x og y er tallene i X- og Y-registrene. (Se side 47).

[Δ%] – Beregner den procentiske forskel mellem tallene i Y- og X-registrene. (Se side 48).

Forvalgstaster

[f] – Bruges til at aktivere de funktioner, der er trykt med gul farve på tastaturpladen over tasterne. (Se side 20).

[g] – Bruges til at aktivere de funktioner, der er trykt med blå farve på tasternes skrå forside. (Se side 20).

CLEAR [PREFIX] – Annullerer brug af [f] og [g] i delvist indtastede instruktioner som f.eks. [f] [SCI] eller [g] [HYP^{-1}]. Samtidig udlæses alle 10 betydende cifre. (Se side 21).

Sandsynlighedsregning

[$P_{y,x}$] – Beregner antallet af

permutationer af *y* ting taget *x* ad gangen. (Se side 50).

[Cy,x] – Beregner antallet af kombinationer af *y* ting taget *x* ad gangen. (Se side 50).

Stakmanipulation

[x≷y] – Ombytter tallene i X- og Y-registrene. (Se side 27).

[R↓] – Ruller stakregistrene ned (f.eks. går y til X). (Se side 27).

[R↑] – Ruller stakregistrene op (f.eks. går x til Y). (Se side 27).

[CLx] – Nulstiller X-registret. (Se side 17).

[←] – I normalstatus bruges [←] til at slette indholdet af lyspanelregistret. Under talindtastning slettes kun det sidst indtastede ciffer, ellers hele indholdet af X-registret. (Se side 17).

Statistik

CLEAR [Σ] – Nulstiller statistikregistrene (R_0 til R_5) samt stakregistrene. (Se side 52).

[Σ+] – Summerer tallene i X- og Y-registrene i summationsregistrene R_0-R_5. (Se side 52).

[Σ-] – Korrigerer fejlindtastninger: Tallene i X- og Y-registrene tages ud af summerne i R_0-R_5. (Se side 55).

[$\bar{x}$] – Beregner middelværdier på basis af tallene i R_0-R_5. (Se side 56).

[s] – Beregner standardafvigelser (stikprøve) på basis af tallene i R_0-R_5. (Se side 57).

[ŷ,r] – Beregner et lineært estimat ($\hat{y}$) for en given *x*-værdi på basis af tallene i R_0-R_5. På basis af samme tal beregnes korrelationskoefficienten *(r)*. (Se side 60).

[L.R.] – Beregner de lineære regressionskoefficienter *(A)* og *(B)* i ligningen $y = Ax + B$. Skæringspunktet på Y-aksen *(B)* lagres i Y-registret, medens hældningskoefficienten *(A)* lagres i X-registret. (Se side 58).

[RAN#] – *(RANdom)* – Beregner et pseudotilfældigt tal på basis af et udgangstal, der er lagret med [STO] [RAN#]. (Se side 51).

Dataregistre

[STO] – *(STOre)* – Lagrer tallet i X-registret i det dataregister, hvis adresse indtastes efter [STO]. (Se side 35).

[RCL] – *(ReCaLl)* – Henter en kopi frem til X-registret af det tal, der er lagret i det dataregister, hvis adresse indtastes efter [RCL]. (Se side 36).

CLEAR [REG] – *(CLEAR REGister)* – Nulstiller alle dataregistre. (Se side 36).

[LSTx] – *(LaST X)* – Henter en kopi fra

SIDSTE X af det foregående indhold af X-registret. (Se side 27).

Trigonometriske funktioner

[DEG] – *(DEGrees)* – Vælger grader som vinkelmål. I lyspanelets symbolrække ses *ingen* indikator. (Se side 43).

[RAD] – *(RADians)* – Vælger radian som vinkelmål. I lyspanelets symbolrække ses RAD. (Se side 43).

[GRD] – *(GRaDes)* – Vælger nygrader som vinkelmål. I lyspanelets symbolrække ses GRAD. (Se side 43).

[SIN], [COS] og [TAN] – Beregner sinus, cosinus og tangens af tallet i X-registret. (Se side 44).

[SIN^{-1}], [COS^{-1}] og [TAN^{-1}] – Beregner invers sinus, invers cosinus og invers tangens af tallet i X-registret. (Se side 44).

Oversigt over programmeringstaster

[P/R] – *(Program/Run)* – Bruges til at skifte mellem programmerings-status (P-status) og normalstatus (R-status). I programmeringsstatus vises PRGM i lyspanelets symbolrække. (Se side 75).

[MEM] – *(MEMory)* – Bruges til udlæsning af den aktuelle splitning. (Se side 73).

[A] [B] [C] [D] [E] – Brugerdefinérbare taster, hvis funktioner svarer til instruktionerne efter den tilhørende [LBL]. (Se side 74).

[USER] – Bruges til at skifte ind og ud af USER-status, hvor øverste rækkes funktioner [$\sqrt{x}$], [e^x], [10^x], [y^x] og [$1/x$] skifter plads med de brugerdefinérbare taster [A] - [E]. (Se side 75).

0 1 2 3 4 5 6 7 8 9 – Tallene kan bruges som etiketadresser, der bruges til at markere starten på et program eller et underprogram. (Se side 74).

[LBL] – *(LaBeL)* – Bruges sammen med [A] - [E] og tallene 0-9 til, som etiket, at markere starten på et program eller et underprogram. (Se side 74).

GTO – *(Go TO)* – Bruges sammen med A - E og tallene 0-9 samt I, der fungerer som adresser ved hop fra den linie, hvor GTO »etiket« er lagret, hen til LBL »etiket«. I normalstatus hopper programpilen hen til den pågældende etiket og standser der. Under programkørsel hopper programpilen hen til etiketten og fortsætter kørslen derfra (uden lagring af returadresse). (Se side 106).

GSB – *(Go to SuBroutine)* – Bruges sammen med A - E og tallene 0-9 samt I, der fungerer som adresser ved underprogramkald fra den linie, hvor »etiket« er lagret, hen til LBL »etiket«. I normaltstatus hopper programpilen hen til den pågældende etiket og starter programudførelsen derfra. Under programkørsel hopper programpilen hen til etiketten og fortsætter kørslen derfra og lagrer samtidig en returadresse, der svarer til linien lige efter GSB »etiket«. (Se side 113).

GTO · *nnn* – *(Go TO line nnn)* – Sætter programpilen ud for programlinie *nnn*. (Se side 74).

BST – *(Back STep)* – Fører programpilen én linie tilbage. (Se side 92).

SST – *(Single STep)* – Fører programlinien én linie frem. I normalstatus udføres den pågældende linie således, at programmet kan udføres linie for linie. I programmeringsstatus udføres instruktionerne ikke. (Se side 92).

← – I programmeringsstatus bruges slettetasten ← til at slette indholdet af en programlinie. Alle efterstående instruktioner rykker en linie op. (Se side 17).

CLEAR PRGM – *CLEAR PRoGraM)* – I programmeringsstatus slettes programhukommelsen. I normalstatus sker ikke andet, end at programpilen føres op til linie 000. (Se side 74).

PSE – *(PauSE)* – Pausefunktion, der stopper programudførelsen i ca. 1½ sekund, medens indholdet af X-registret udlæses. (Se side 75).

R/S – *(Run/Stop)* – Bruges til at skifte mellem programudførelse og normalstatus. (Se side 75).

RTN – *(ReTurN)* – Bruges til at afslutte et program eller et underprogram. Hvis RTN afslutter et kaldt underprogram, føres programpilen tilbage til linien efter

underprogramkaldet, ellers føres pilen til linie 000, hvor kørslen stopper. (Se side 75).

[SF] – *(Set Flag)* – Bruges til at *sætte* flag 0 eller 1. (Se side 106).

[CF] – *(Clear Flag)* – Bruges til at *slette* flag 0 eller 1. (Se side 106).

[F?] – Bruges til at *teste* et flag. Hvis flaget er sat, udføres den efterfølgende programlinie i hukommelsen, ellers springes den over. (Se side 106).

[x≤y], [x>y], [x≠y] og [x=y] – Sammenligningsoperationer, hvor indholdet af X-registret sammenlignes med indholdet af Y-registret. Hvis betingelsen er opfyldt, udføres den efterfølgende programlinie, ellers springes den over. (Se side 105).

[x<0], [x>0], [x=0] og [x≠0] – Sammenligningsoperationer, hvor indholdet af X-registret sammenlignes med 0. Hvis betingelsen er opfyldt, udføres den efterfølgende programlinie, ellers springes den over. (Se side 105).

Stikordsregister

Sidetal i **halvfede** typer indikerer primær reference.

A

Adresse
  dataregister, **35-39**
  etiket, **74,** 89
  indirekte (se Indirekte adressering)
Afrundingsfejl, **68,** 147, 161
  statistik, **55**
Afslutning på indtastning, 17, 18, **231**
Afslutning på program, **79**
Alfaetiketter, **74**
Algoritme, **200-202**
Alternative tastfunktioner, **20-21,** 75
Andengradsligning, **200-204**
Annuiteter (se Finansregning)
Automatisk rullestak (se Rullestak)

B

Back Step, **92,** 95, 98
Baktast, **17,** 28, 93
Batterier, **217-220**
  installation, **218-220**
  typer, **218**
  udskiftning, **218**
Blokering af stakløft, 28, **231**
Brugerdefinérbare taster, 206, **211-212,** 237

C

Chi-i-anden-test, **190-193**
Cifferinddeling, **16**
Cirkelareal, **79 ff**

D

Dataregistre, 9, **35-36, 70-73**
  adresser, 9, **35-38**
  aritmetik, **37,** 236-237
  konvertible, **70-73**
  nulstilling, **36**
  statistik, **52-53**

Decimaltegn, 16
Dekrementering, **122-123,** 207, 236
Determinant
  matrix-algebra, **132-140**
Lineære ligningssystemer, **141**
DSE-grænser, **128**
Dåsevolumen, **84-86**

E

Eksponent
  fortegn, 64, 66, 67
  grænser, **67**
Eksponentiel kurve, 154, **158-159**
Etiket, **74,** 79, 89
  adresse (se Adresse)
  søgning efter, **82**

F

Fejl
  afrundingsfejl, **68,** 161
  fejlmeddelse, 18, 88, 91, **213-215**
  flag, **209**
  kontinuerlig hukommelse, **19,** 215, 220
  ophævelse af, **19**
  service, 215, 221, **222-224**
  Simpson's metode, **194-197**
  statistikregistre, **214,** 229
  underprogram, **115,** 215
Fejlsøgning, **220-222**

Finansregning, **175-184**
Flag, 19, **106,** 109-112, 235, 209-210
Forgrening
  ubetinget, **105**
  ubetinget, **106**
Forkortet indtastning, **74,** 121
Fortegn
  eksponent, 64, **66**
  mantisse, **17,** 67
Fortegnsvender, **17,** 41, 232
  eksponent, **66**
Forvalgstast, **20,** 74, 75, 121
  sletning, **21**
Fremkald af tal, **36**
  stakløft, **36**
summation med (Σ +), **52**
Funktioner af én variabel, **21,** 24, 42-47
Funktioner af to variable, **21,** 24, 28, 47-51
  summation, **52**

G

Goto, **92-93, 107, 113-114, 125**

H

Hop
  etiket, **106**
  indirekte, **125,** 129-130, 235
  linie, editering, **92-96**
  underprogram, matrix-algebra, **204**
  underprogram, **113-114,** 204

I

Ikke-programmérbare funktioner, **83**
Indeksregister, 9, 71-72, 76, **121ff**
  dekrementering, **122-124**
  direkte funktioner, **121-124**
  inkrementering, **122-124**
  lagring og fremkald, **121**
  ombytning, **121**

testværdi til løkker, **123**
Indikatorer, **16**
f, 20
g, 20
PRGM, 19, **74, 76,** 80, 83
USER, 75, 83
Indirekte adressering, 121, **124,** 205, 206
forgrening, **125,** 129
fremkald og lagring, **124,** 128, 236
linienummer, **125,** 130
ombytning, **124,** 128, 236
registeraritmetik, **124,** 128
tabel, **125**
underprogrammer, **125,** 129
Indledning af program, **79**
Indlæsning af data, **206-207,** 211
Indlæsningsrutiner, **235**
Indtastning, **22-24**
Indtastning af program, **80-81**
Initialisering, **91**
Inkrementering, **122-123,** 126-128, 207, 236
grænser, **128**
indirekte adressering, **205-206**
Intern præcision, **63,** 67

K

Konstantaritmetik, **33-35**
Kontinuerlig hukommelse, **19,** 71, 74, 91
fejl, **20,** 215, 220
flag, 19, **106**
lyspanel, **63**
nulstilling, **20,** 36, 44, 51, 71, 215, 220
tab af, **218,** 219
tilfældige tal, **51**
udskiftning af batterier, 20, **218-219**
vinkelmål, **44**
Konvertible dataregistre, 9, **70-73**
Korrektionstast, **17,** 28

Korrelationskoefficient, **155**
Kurvetilpasning, **154-159**
  program, falg, **209-210**
Kæderegning, **30-31**
Kørestatus, **19**
  tast, **75**
running, 18, 51, 83
Kørsel af program, **81-83**

L

Lager
  automatisk splitning, **70-73,** 225-228
  konvertering (se Splitning)
  konvertering til dataregistre, **228-229**
  MEM-funktionen, **73,** 228-229
  splitning, **73**
Lagring af tal, **35**
Ligningssystem, **141-145**
Lineær regression, **154,** 236
Lineært ligningssystem, **132-140**
Linie 000, **76**
Linienummer, **73-74**
Logaritmiske kurver, 154, **159**
Lyspanel
  afrunding, **63-66**
  automatisk skift mellem FIX og SCI, **64**
  indikatorer, **16**
  lyspanelregister, **25**
  nulstilling/sletning, **17,** 28
  status, 19, 63
  tomt, **220-221**
  udlæsningsformat, 18, **63**
  X-register, **25**
Løkker, **107-108,** 207-209, 236
  betingelser ved Newton-Raphson iteration, **161**
  betinget kontrol, **108-109**
  grænse for $\triangle x$, **207-208**
  kontroltal, **122**

atricer, ligninger, **139**
tæller, 121, **122ff,** 126-128, 207
udhop, **107-109,** 121, 122-124, 126-128, 207-209
Løkketæller, **207**

M

Mantisse, 21, 63, 64, 66, **67**
Matricer
indirekte adressering, **205**
inkrementering, **205**
underprogrammer, **204**
Mellemresultat, **22-24,** 25, 30
Mindste kvadraters metode, **156**
Modificering af program, **96-101**

N

Negative tal, **17**
Neutral stak, **232-233**
løkker, **207**
Newton-Raphson-iteration, **160-164**
Nulstilling
dataregistre, **36**
kontinuerlig hukommelse, **20,** 36, 44, 51, 71, 215, 220
lyspanel, **17-18**
rullestak, **52**
statistikregistre, **52,** 54
Nulstilling af kontinuerlig hukommelse,
36, 44, 51, 71, 215, 220
Numerisk integration, 194-197
Numeriske etiketter, **74**
Nøjagtighed (se Afrundingsfejl)

O

Omnummerering af programinstruktioner, **96-97**
(ON), 11, **16,** 220
Ophævelse af fejltilstand, **19**
Overløb, **18,** 91, 213

P

Permanent dataregister, 9, **71,** 225
Permanent programlager, **71,** 225-226
Potenskurver, **154**
PRGM-indikator, 19, **73,** 76, 80
Primære funktioner, **20,** 75
Problemformulering, **199**
Program
  etiketter, **89**
  fejl, **91**
indsættelse af instruktioner, **96,** 99
  kontroltaster, **242-244**
  modificering, **96-101**
  pauser, **86-87**
  planlægning, **199-204**
  selvinitialisering, **91**
  sletteinstruktioner, 17, **93,** 99-101
  status, **19**
  stop, 75, **83-88**
  struktur, **199-202**
  taster, **74**
  udførelse, **82,** 109, 112
Programhukommelse, 9, **74,** 76-77
  afsat lagerplads, **79,** 82, 95, 99
  linie 000, **76**
  nulstilling, **74**
  sidste programlinie, **79**
  sletning af instruktioner, **99-101**
  sletning, **74**
Programlager (se Programhukommelse)
Programmeringsteknik, Indeks, **235-237**
Programpil
  linievis fremføring, **92,** 94-96
  linievis programkørsel, **94-95**
  linievis tilbageføring, **92,** 95, 98
Pseudotilfældige tal (se Tilfældige tal)
Pythagoras, **93ff**

R

R/S-indlæsning, **207**
Rul-ned, **27**
Rul-op, **27**
Rullestak, **9,** 76
  (≥+), **53**
  (RCL) (≥+), **54**
  (s), **57-58**
  ($\bar{x}$), **56-57**
  ($\hat{y}$,r), **60-61**
  aktivering, 28, **231-232**
  automatisk rullestak, **25-26**
  blokering, **21,** 28, 36
  fald, **26-27,** 29, 30, 31
  hukommelse, **231-233**
  ingen ændring, **25-26**
  lineær agression, **58-59**
  manipulation, **26-27**
  neutral, 231, **232-233**
  nulstilling, 36, **52,** 53
  procent, **47-48**
Rutediagram, 110, **202-204,** 210

S

Sammensat rente (se Finansregning)
Sand/Falsk-test, **105**
Selvtest, **220-221**
Service, 220, **223-224**
  forsendelse, **223**
  udbedring af mangler, **222**
SIDSTE X, 9, 27, **31-32,** 76, 233
  estimater og korrelation, **60-61**
  konstantaritmetik, **33-34**
  nulstilling, **36**
  procent, **47-48**
  procentisk forskel, **48**
  sandsynlighedsregning, **50-51**

statistik, **31-32,** 61
summation med ≥, **53**
(≥+), **52-53**
Simpson's metode, **194**
Sletning
fejlmeddelelser, **18**
programhukommelse, **74**
programlinie, **93**
Sletning af programlinier, **99-101**
Splitning, 9, **71**
aktuel, **73**
Single Step, **92,** 95-97
Stakløft
aktivering, 28, **231-232**
blokering, 28, **231**
Statistik
dataregistre, **52-53**
fejl, **214,** 229
korrektion, **55-56**
nulstilling af registre, **52,** 54
præcision, **53-54**
registre, 225, **229**
Status
lyspanel, **63-66**
vinkelmål, **43-44**
Stop af program, **75,** 83-84, 88
Strøm
forbrug, **217**
lav batterispænding, **218**
strømsvigt, 44, **215**
Strømsløjfer, **139-140**

T

T-registret som konstantregister, **33-34**
t-statistik, **184-190**
Talindtastning
afslutning af, **17,** 18, 231
nyt tal, **17,** 24

Tastindeks
funktioner, **238-242**
programmeringstaster, **242-244**
Tastkoder, **73,** 77-78
Temperaturspecifikationer, 223
Test af HP-11C, **220-221**
Test af programmer, **91**
Tilfældige tal, **51-52,** 210-211, 236
Toptaster (tasterne i øverste række), **75-76**
Trapez-metoden, **194-196**
Trekantsløsninger, **146-154**
Indlæsning af data, **206-207**

U

U-bådsjagt, **164-175**
Ubetinget forgrening, 106-107
Udførelse af etiketter, **74-75**
Udførelsestid, **51**
Udgangstal (tilfældige tal), **51-52,** 165, 210-211
Udlæsningsrutiner, **236**
Underløb, **18,** 88
Underprogram, **204-205,** 237
grænser, **114-115**
hop, **113-114, 115ff**
niveauer, **114-115**
ventende returhop, 19, **204-205**
USER-indikator, **75-76,** 83
USER-status, **75-76,** 83

V

Variable, indlæsning, **206-207**
Varmetabsprogram, 12, **76ff**
Ventende returhop, 19, **113-114**
Vinkelmål, 19, **43,** 147
Ændring af udlæst tal, **41-42**

1 Grundbegreberne

2 Den automatiske rullestak, SIDSTE X og dataregistre

3 Matematiske funktioner

4 Udlæsningsformater

5 Grundlæggende programmering

6 Editering

7 Forgreninger og kontroloperationer

8 Underprogrammer

9 Indeksregistret

10 Programsamling

11 Programmeringsteknik

HEWLETT PACKARD

Hewlett-Packard A/S:
Datavej 52, 3460 Birkerød, tel. (02) 81 66 40

Hewlett-Packard Norge A/S:
Østerndalen 18, 1345 Østerås, tel. (02) 17 11 80

00011-90018 REV.B. CP

Printed in Singapore