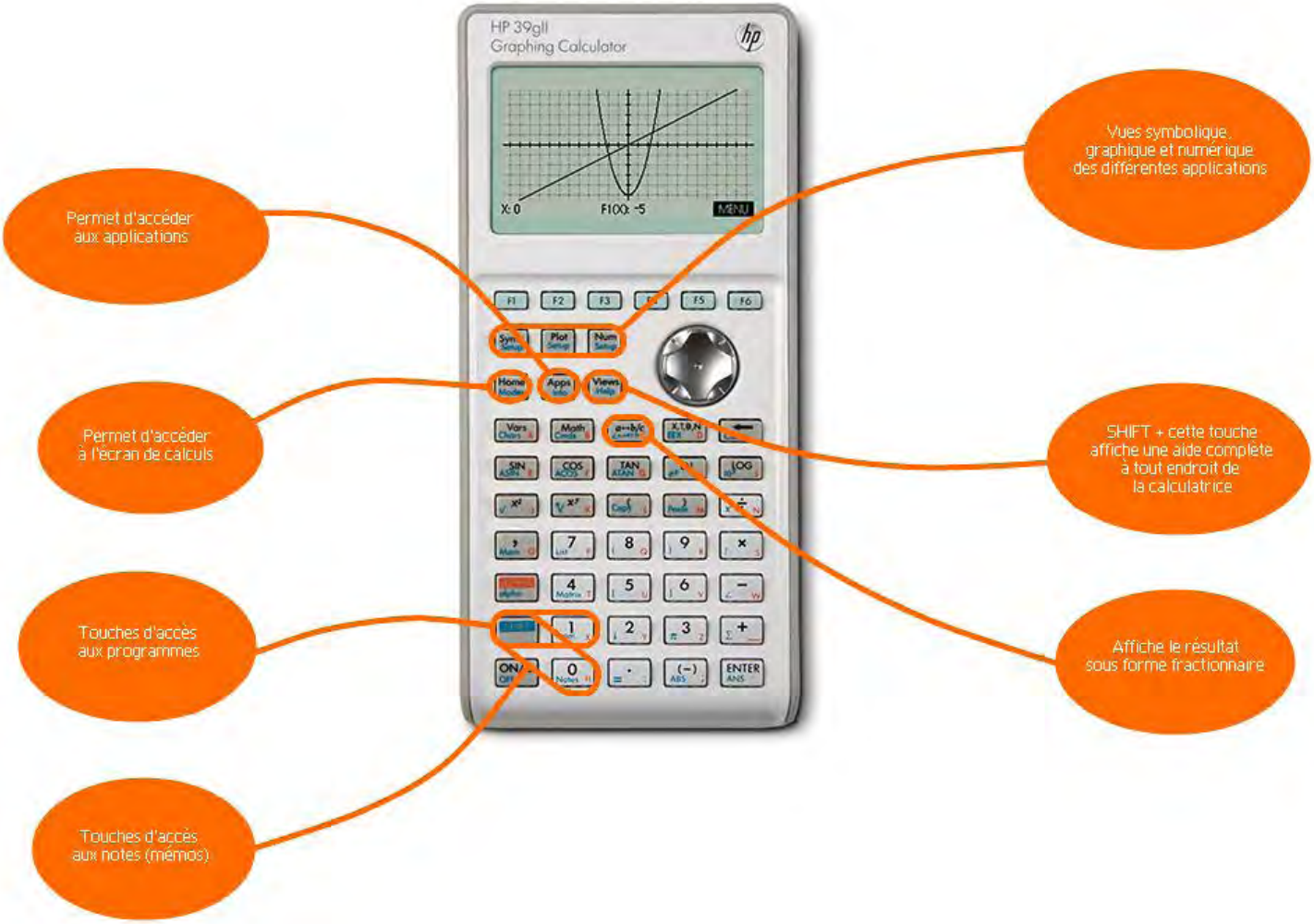
TUTORIAUX

HP 39gII

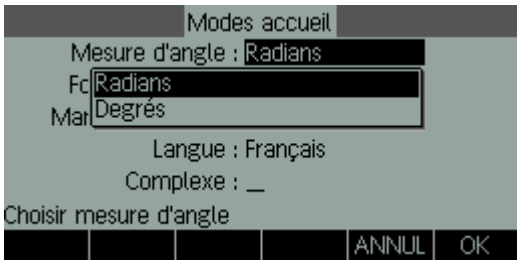
Sommaire

1ère & Tle S

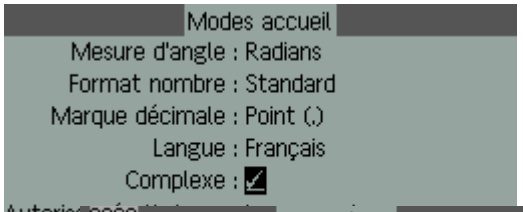
CALCULATRICE HP 39gII



- **Pour allumer la calculatrice :** Taper sur la touche ON/C OFF.
- **Pour éteindre la calculatrice :** Taper sur la touche SHIFT, puis sur la touche ON/C OFF.
- **Pour choisir le mode « degré » :**
  - Ouvrir la fenêtre de configuration en tapant SHIFT Home Modes.
  - Choisir *Degrés* ou *Radians* depuis F2 (CHOIX).



- **Pour activer le mode complexe :**
  - Descendre dans le menu et cocher *Complexe* avec F2.



- **Pour accéder aux commandes de la calculatrice :**
  - Toutes les commandes sont regroupées dans le catalogue accessible depuis les touches SHIFT Math Cmds.

- **Pour accéder aux caractères spéciaux :**
  - La calculatrice possède un nombre impressionnant de caractères spéciaux accessibles depuis les touches SHIFT Vars Chars.

La boîte sans couvercle

HP 39gII

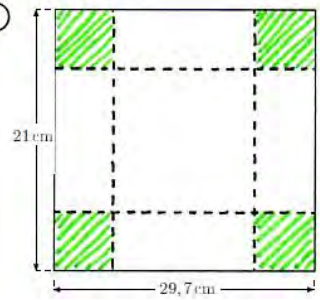
2nde

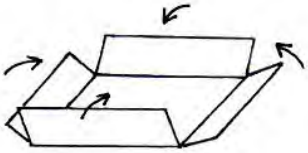
Niveau : 2nde

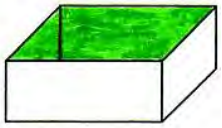
Objectifs : introduction à la notion de fonction et ses représentations. Notion de maximum.
Mots-clés : fonctions, tableau de valeurs, représentation graphique, maximum.

Enoncé : On construit une boîte sans couvercle à l'aide d'une feuille A4 en découpant aux quatre coins des carrés identiques. On cherche la longueur des côtés de ces carrés pour que le volume de la boîte soit le plus grand possible.

① ② ③

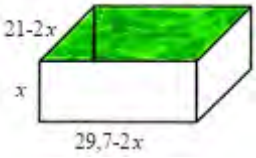






Solution pas à pas :

Chaque élève construit sa boîte et compare son volume avec les autres élèves.
On peut établir un tableau de valeurs regroupant les volumes obtenus en fonction du côté des carrés choisi.
Avec x la longueur du côté des carrés, le volume $V(x)$ de la boîte s'obtient avec ce calcul :



$V(x)$ = Hauteur x longueur x largeur

$V(x) = x\,(29{,}7 - 2x)(21 - 2x)$

Sur la calculatrice HP-39gII, on accède à l'application *Fonction* depuis la touche [Apps Info].

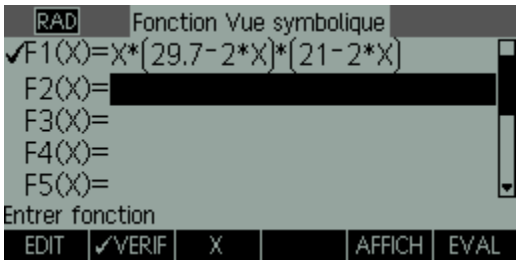
Captures d'écran :

La boîte sans couvercle

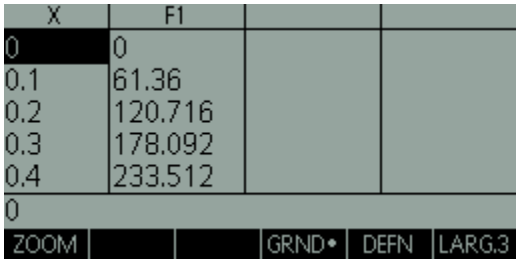
HP 39gII

2nde

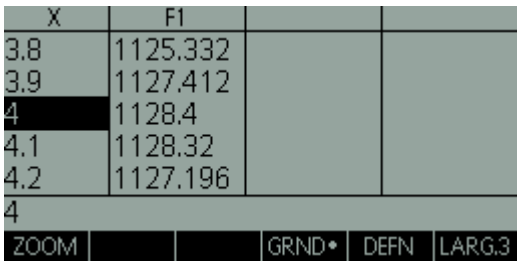
On entre alors l'expression fonction de x après F1(X)=



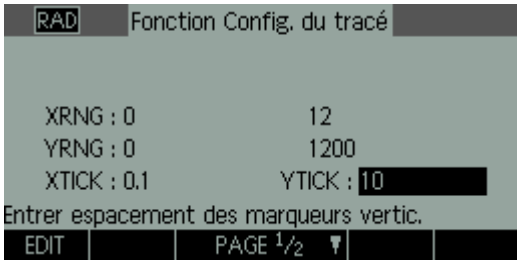
On obtient un tableau de valeurs en appuyant sur la touche Num Setup .



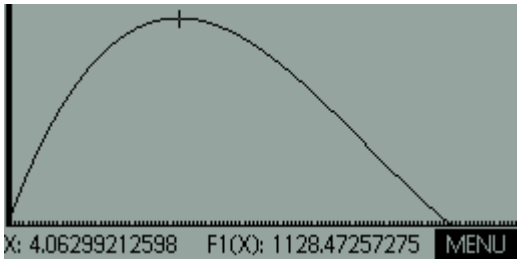
On observe une diminution du volume de la boîte après x = 4.



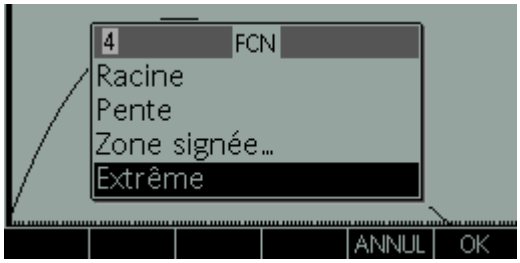
Avant d'obtenir la représentation graphique, il faut régler la fenêtre de tracé en établissant les valeurs minimales et maximales des abscisses (x) et des ordonnées (V(x)). Pour cela, il faut appuyer sur les touches SHIFT et Plot Setup .



La touche Plot Setup permet d'obtenir le tracé. On peut lire une valeur approchée de x quand V(x) est maximal.



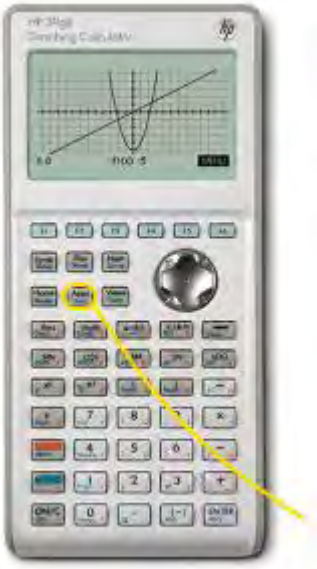
En appuyant sur F6 (MENU) > FNCT (F4) et en choisissant Extrême, la calculatrice nous positionne directement sur le maximum.



Etude de fonctions

HP 39gII

2nde



Niveau : 2nde

Exercice type : Soit f la fonction définie sur [-4 ; 2] qui à x associe $f(x) = \frac{2x+2}{x+5}$

1/ Editer un tableau de valeurs de f ;

2/ Tracer la courbe représentative de f.

Touche d'accès aux fonctions :

Apps Info

Solution pas à pas :

Accéder aux applications depuis la touche Apps Info .

Aller sur Fonction avec les touches fléchées et appuyer sur F6 .

Entrer la forme algébrique de la fonction avec cette séquence de touches :

(Copy L) 2 i Y X,T,θ,N EEX D + Σ _ 2 i Y) Paste M

x^{-1} ÷ N (Copy L) X,T,θ,N EEX D + Σ _ 5 [U) Paste M

ENTER ANS

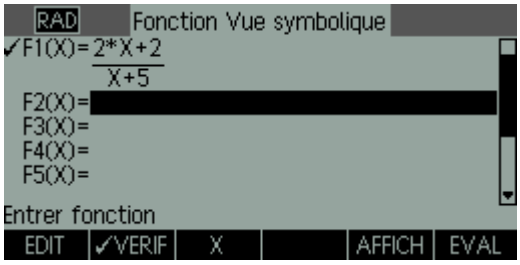
Puis valider avec F6 .

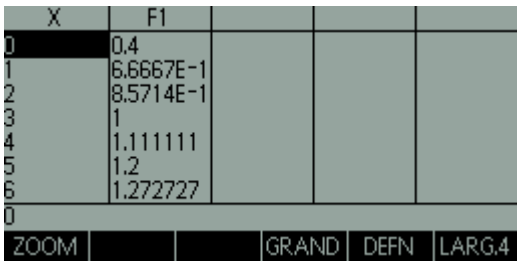
Appuyer sur la touche Num Setup pour directement accéder au tableau de valeurs.

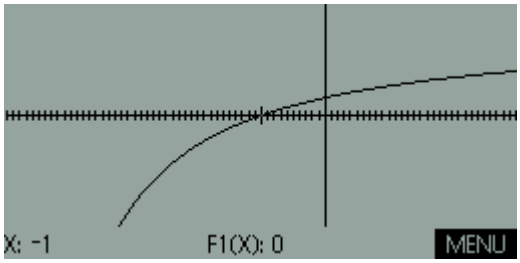
Appuyer sur la touche Plot Setup pour directement obtenir la représentation graphique.

Astuce : appuyer sur les touches + Σ _ ou - ∠ W pour directement zoomer ou reculer.

Captures d'écran :



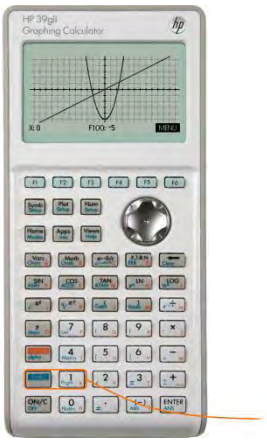




Premiers algorithmes & boucles

HP 39gII

2nde



Niveau : 2nde
Objectifs : L'algorithmique fait maintenant partie intégrante des programmes de mathématiques au lycée.
Dès la seconde, les élèves découvrent différents algorithmes.
Voici une sélection d'algorithmes rencontrés au lycée.

Touches d'accès aux programmes :

Solution pas à pas :

Premier exemple : premier algorithme :

Écrire un algorithme qui demande d'entrer un nombre puis affiche son image par la fonction *f* définie par
f(x) = x^2 + 6x - 4.

Algorithme
Entrée
Demander à l'utilisateur l'antécédent *x*
Traitement
Affecter *x*^2+6**x*-4 à la variable *y*
Sortie
Afficher *y*

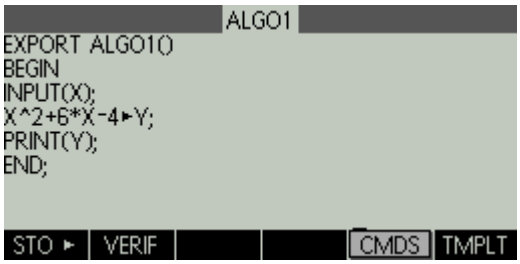
Deuxième exemple : boucle « Pour » :

Écrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

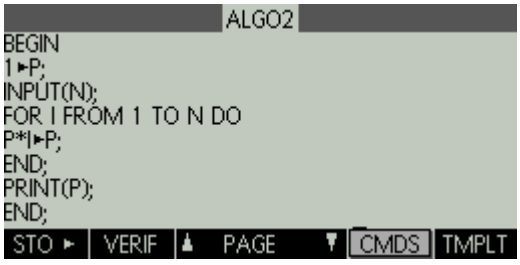
Algorithme
Entrée
Demander à l'utilisateur un nombre de départ *n*
Initialisation
Nombre *p* initialisé à la valeur 1
Traitement
Pour *i* allant de 1 à *n*
Stocker *p***i* dans *p*
Fin de la boucle pour
Sortie
Afficher *p*

Captures d'écran :

Sur HP-39gII, on écrira :



Sur HP-39gII, on écrira :



Premiers algorithmes & boucles

HP 39gII

2nde

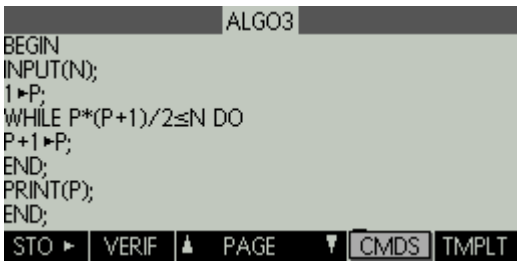
Troisième exemple : boucle « Tant que » :
Trouver le plus petit entier *p* tel que la somme des entiers de 1 à *p* soit inférieure à un entier *n* donné.

On rappellera la formule (1ère ES / S) :

$$\sum_{k=1}^{p} k = \frac{p(p+1)}{2}$$

Algorithme
Entrée
Demander à l'utilisateur un nombre *n*
Initialisation
Nombre *p* initialisé à la valeur 1
Traitement
Tant que p*(p+1)/2 est inférieure à n
Stocker *p*+1 dans *p*
Fin de la boucle tant que
Sortie
Afficher

Sur HP-39gII, on écrira :



Algorithme : formule de Héron

HP 39gII

La formule de Héron permet de calculer l'aire d'un triangle :

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

avec p le demi-périmètre du triangle.

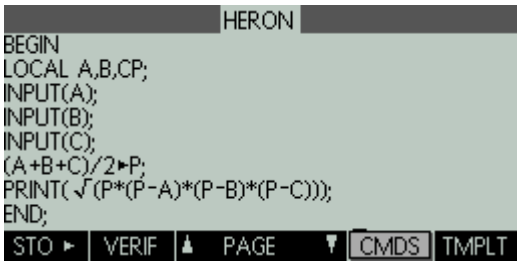
Programmer un algorithme donnant l'aire d'un triangle avec la formule de Héron.

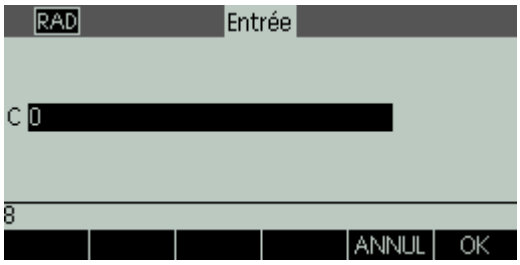
Solution pas à pas :

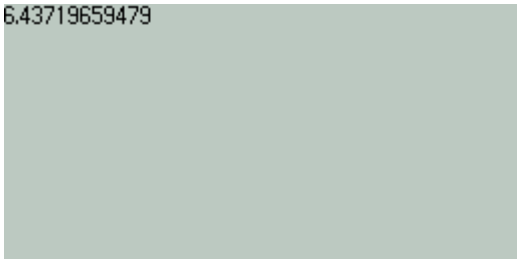
On crée un programme HERON depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT HERON()
BEGIN
LOCAL A,B,C,P;
//On demande à l'utilisateur les 3 longueurs du triangle
INPUT(A);
INPUT(B);
INPUT(C);
// On calcule le périmètre du triangle
(A+B+C)/2▸P ;
//On calcule l'aire avec la formule de Héron
PRINT(√(P*(P−A)*(P−B)*(P−C));
END;
```

Captures d'écran :







Algorithme : calcul de l'IMC

HP 39gII

L'IMC (indice de masse corporelle) évalue la santé pondérale (corpulence). Il permet notamment de mettre en évidence le surpoids ou l'obésité.
Le calcul de l'IMC n'est qu'un critère indicatif car la masse osseuse et musculaire n'est pas prise en compte.
L'IMC se calcule avec cette formule :

$$\frac{P}{T^2}$$

avec P la masse en kilogrammes et T la taille en mètres.

L'OMS a établi cette classification :

Classification OMS	IMC (kg/m²)
Déficit pondéral	<18,5
Poids normal	18,5 – 24,9
Surpoids	25 – 29,9
Obésité modérée (classe I)	30 – 34,9
Obésité sévère (classe II)	35 – 39,9
Obésité morbide (classe III)	≥ 40

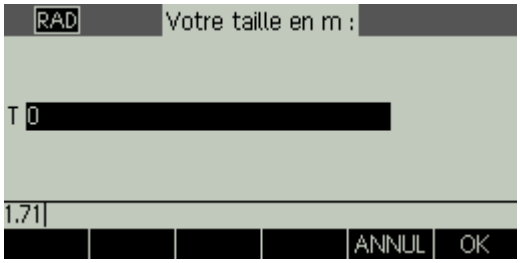
Créer un algorithme calculant l'IMC et donnant la classification OMS.

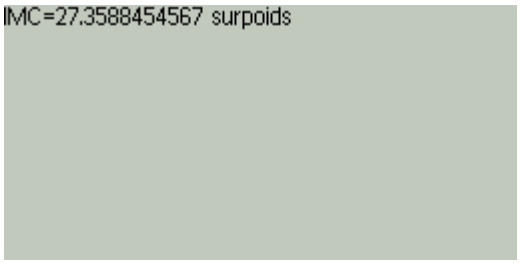
Solution pas à pas :

On crée un programme IMC depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT IMC()
BEGIN
LOCAL P,T,I;
//On demande à l'utilisateur son poids et sa taille
INPUT(P,"Votre poids (masse) en kg :");
INPUT(T,"Votre taille en m :");
// On calcule l'IMC
P/T^2▸I ;
//On classe
IF I<18.5 THEN PRINT("IMC="+I+" déficit pondéral"); END;
IF I≥18.5 AND 24.9≥I THEN PRINT("IMC="+I+" poids
normal"); END;
IF I≥25 AND 29.9≥I THEN PRINT("IMC="+I+" surpoids");
END;
IF I≥30 AND 34.9≥I THEN PRINT("IMC="+I+" obésité
modérée (classe I)"); END;
IF I≥35 AND 39.9≥I THEN PRINT("IMC="+I+" obésité sévère
(classe II)"); END;
IF I≥40 THEN PRINT("IMC="+I+" obésité morbide (classe
III)"); END;
END;
```

Captures d'écran :





Algorithme : jeu du nombre mystère

2nde

HP 39gII

Programmer un algorithme où l'utilisateur doit trouver un nombre entier tiré aléatoirement entre 1 et 100 et où est précisé, à chaque essai, si le nombre saisi est supérieur ou inférieur au nombre mystère.

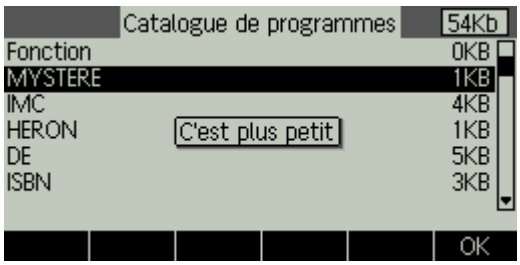
Solution pas à pas :

On crée un programme MYSTERE depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT MYSTERE()
BEGIN
LOCAL M,N;
//On tire aléatoirement un nombre entier entre 1 et 100
1+FLOOR(100*RANDOM)▸N;
//On demande à l'utilisateur de saisir un nombre
INPUT(M);
//On redemande à l'utilisateur de saisir un nombre tant qu'il ne
correspond pas au nombre mystère en précisant avant si le
nombre précédent est inférieur ou supérieur au nombre
mystère
WHILE M<>N DO
  IF M>N THEN
    MSGBOX("C'est plus petit") ;
  ELSE
    MSGBOX("C'est plus grand") ;
  END;
  INPUT(M);
END;
MSGBOX("Nombre mystère trouvé ! ");
END;
```

La commande MSGBOX est semblable à PRINT sauf qu'elle affiche le texte dans une boîte de dialogue et non sur l'écran de sortie.

Captures d'écran :



Calcul de PGCD par soustractions

HP 39gII

2nde

Programmer un algorithme affichant les étapes de calculs d'un PGCD avec la méthode des soustractions.

Solution pas à pas :

On crée un programme SOUST depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT SOUST()
BEGIN
LOCAL A,B,C;
//On demande à l'utilisateur deux nombres entiers nuls dont on veut calculer le PGCD
INPUT(A);
INPUT(B);
PRINT(A+" ; "+B);
//On prend le plus petit des 2 et la soustraction du plus grand et du plus petit
MIN(A,B)▸C;
MAX(A,B)−MIN(A,B)▸B;
C▸A;
PRINT(A+" ; "+B);
//On reprend à nouveau le plus petit et la différence tant qu'on n'obtient pas la même chose
WHILE A<>B DO
  MIN(A,B)▸C;
  MAX(A,B)−MIN(A,B)▸B;
  C▸A;
  PRINT(A+" ; "+B);
END;
//On affiche la valeur du PGCD
PRINT(C) ;
END;
```

Captures d'écran :

Calcul de PGCD par la méthode d'Euclide

2nde

HP 39gII

Programmer un algorithme affichant les étapes de calculs d'un PGCD avec la méthode d'Euclide.

Solution pas à pas :

On crée un programme EUC depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT EUC()
BEGIN
LOCAL A,B,C;
//On demande à l'utilisateur deux nombres entiers nuls dont on veut calculer le PGCD
INPUT(A);
INPUT(B);
PRINT(A+" ; "+B);
//On prend le plus petit des 2 et le reste dans la division euclidienne du plus grand par le plus petit
MIN(A,B)▸C;
irem(MAX(A,B),MIN(A,B))▸B;
C▸A;
PRINT(A+" ; "+B);
//On reprend à nouveau le plus petit et le reste tant qu'il n'est pas nul
WHILE B<>0 DO
  MIN(A,B)▸C;
  irem(MAX(A,B),MIN(A,B))▸B;
  C▸A;
  PRINT(A+" ; "+B);
END;
//On affiche la valeur du PGCD
PRINT(C) ;
END;
```

Captures d'écran :

Algorithme : tour de magie

HP 39gII

2nde

Un magicien demande à un spectateur :

- De penser à un nombre ;
- De prendre son double ;
- D'enlever 3 ;
- De faire une multiplication par 6 ;
- D'annoncer le résultat obtenu.

Ecrire un programme SPECT qui affiche le nombre annoncé au magicien par le spectateur et un programme MAGIE qui retrouve le nombre pensé par le spectateur à partir du résultat annoncé.

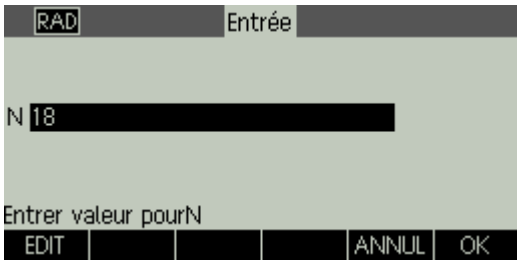
Solution pas à pas :

On crée un programme EUC depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT SPECT()
BEGIN
LOCAL N;
//On demande au spectateur d'entrer le nombre auquel il pense
INPUT(N);
//On effectue les calculs demandés par le magicien et on l'affiche
PRINT((2*N−3)*6);
END;

EXPORT MAGIE()
BEGIN
LOCAL N;
//On entre le nombre annoncé par le spectateur
INPUT(N);
//On remonte le programme de calculs en faisant les opérations contraires et on affiche le résultat qui est le nombre pensé par le spectateur
PRINT(((N/6+3)/2));
END;
```

Par exemple, le spectateur pense à 18.
Il annonce alors 198.
Le programme MAGIE retourne bien 18 avec 198 comme entrée.

Captures d'écran :



Algorithme : année bissextile

HP 39gII

2nde

Les années bissextiles sont les années qui sont :

- Soit divisibles par 4 mais non divisibles par 100 ;
- Soit divisibles par 400.

Ecrire un programme indiquant si une année est bissextile.

Solution pas à pas :

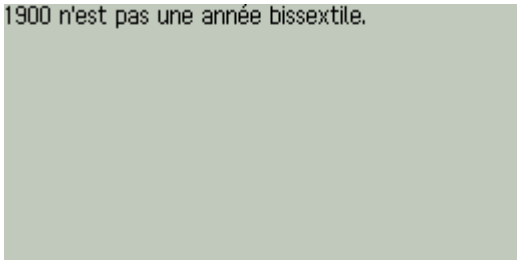
On crée un programme EUC depuis l'éditeur (touches SHIFT 1 Prgm x) et on tape l'algorithme suivant :

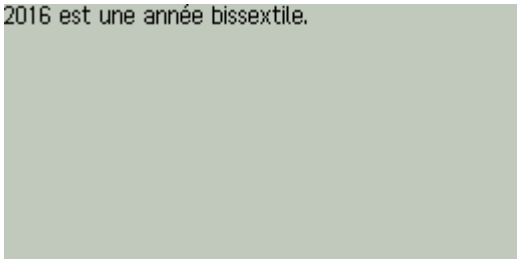
```
EXPORT BISS()
BEGIN
LOCAL N;
//On demande à l'utilisateur d'entrer l'année
INPUT(N);
//On vérifie les conditions sur l'année bissextile
IF (irem(N,4)==0 AND irem(N,100)<>0) OR irem(N,400)==0
THEN
  PRINT(N+" est une année bissextile.");
ELSE
  PRINT(N+" n'est pas une année bissextile.");
END;
END;
```

Pour utiliser l'algorithme suivant « Quel jour êtes-vous né ? », on place directement l'entrée dans le nom du programme et on remplace la sortie par 1 si l'année est bissextile et 0 sinon :

```
EXPORT BISS(N)
BEGIN
IF (irem(N,4)==0 AND irem(N,100)<>0) OR irem(N,400)==0
THEN
  RETURN(1);
ELSE
  RETURN(0);
END;
END;
```

Captures d'écran :





Algorithme : quel jour êtes-vous né ?

HP 39gII

2nde

Voici une méthode pour déterminer le jour de la semaine d'une date donnée comprise entre 1900 et 2099 :

- On associe à chaque mois un code en utilisant le nombre 033 614 625 035 (janvier = 0, février = 3, etc) ;
- On additionne : le nombre formé des deux derniers chiffres de l'année, le quart de ce nombre (tronqué à la virgule si ce n'est pas un entier), la date du jour (donc un entier entre 1 et 31), le code du mois.
- Si la date est après 2000, on enlève 1 au résultat ;
- Si l'année est bissextile et si la date est avant le 1er Mars, on enlève 1 au résultat ;
- On divise par 7, et le reste donne le jour de la semaine (0 = dimanche, 1 = lundi, etc).

Ecrire un programme qui retourne le jour d'une date de naissance.

Solution pas à pas :

On crée un programme JOUR depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT JOUR()
BEGIN
LOCAL A,M,J,N,P,L1,L2;
//On demande à l'utilisateur d'entrer sa date de naissance
//On demande à l'utilisateur d'entrer l'année
INPUT(A,"Année ?");
//On demande à l'utilisateur d'entrer le mois
INPUT(M,"Mois (de 1 à 12) ?");
//On demande à l'utilisateur d'entrer le jour
INPUT(J,"Jour (de 1 à  31) ?");
//On crée une liste contenant le code des mois
{0,3,3,6,1,4,6,2,5,0,3,5}▸L1;
//On regarde si l'année est après 2000 pour retirer 1
0▸P;
IF A>2000 THEN P-1▸P; END;
//On regarde si l'année est bissextile et le mois avant mars
pour retirer 1
IF BISS(A)==1 AND M<3 THEN P-1▸P; END;
//On extrait les deux derniers chiffres de l'année
irem(A,100)▸A;
//On effectue le calcul décrit dans le sujet
A+FLOOR(A/4)+J+L1(M)+P▸N;
//On effectue la division par 7 pour obtenir le jour
{"dimanche","lundi","mardi","mercredi","jeudi","vendredi","same
di"}▸L2;
irem(N,7)▸N;
PRINT("Tu es né un "+L2(N+1));
END;
```

Captures d'écran :

Algorithme : tirage graphique d'un dé

HP 39gII

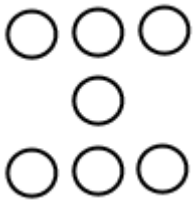
2nde

Simuler graphiquement le tirage d'un dé à 6 faces.

Solution pas à pas :

Pour tirer un nombre entier aléatoire entre 1 et 6, on utilise la commande RANDOM qui tire un nombre décimal compris entre 0 et 1 avec 12 chiffres après la virgule. L'astuce est de prendre la partie entière ; avec la commande FLOOR(; du produit de 6 par RANDOM et d'ajouter 1 : 1+FLOOR(6*RANDOM)

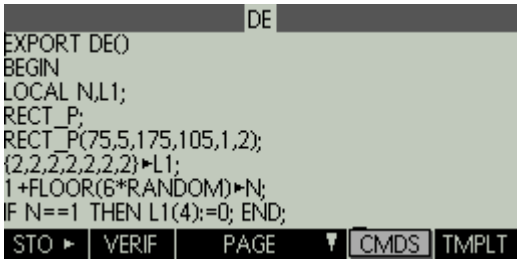
Pour simuler graphiquement le dé, on affiche un carré avec la commande graphique RECT_P puis on remplit avec la commande ARC_P les 6 cercles suivants selon le chiffre tiré aléatoirement :

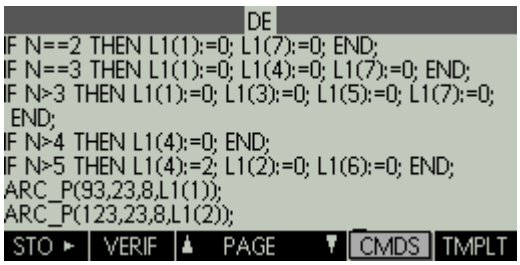


On programme alors cet algorithme sur la HP 39gII :

```
EXPORT DE()
BEGIN
LOCAL N,L1;
//On efface l'écran et on dessine le carré de la face du dé
// Le carré est remplit de gris clair (couleur 2)
RECT_P ;
RECT_P(75,5,175,105,1,2) ;
//On règle la couleur des 7 cercles à gris clair (couleur 2) pour
//qu'ils soient invisibles sur la face : cercles éteints
{2,2,2,2,2,2,2}▸L1 ;
//On tire aléatoirement un nombre entre 1 et 6
1+FLOOR(6*RANDOM)▸N;
//On traite l'affichage dans les 6 cas où on allume les cercles
//correspondants en noir (couleur 0)
IF N==1 THEN L1(4):=0; END;
```

Captures d'écran :





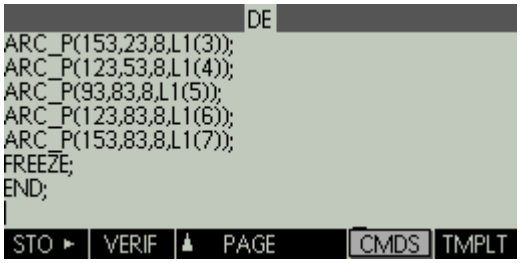
Tirage graphique d'un dé

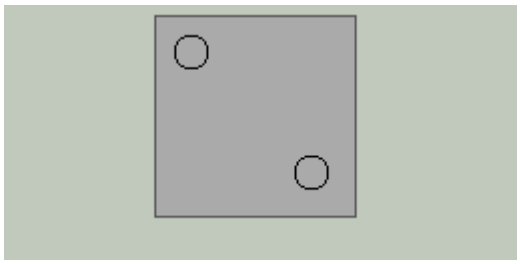
HP 39gII

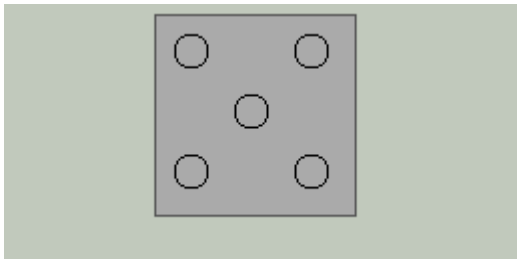
2nde

```
IF N==2 THEN L1(1):=0; L1(7):=0; END;
IF N==3 THEN L1(1):=0; L1(4):=0; L1(7):=0; END;
IF N>3 THEN L1(1):=0; L1(3):=0; L1(5):=0; L1(7):=0; END;
IF N>4 THEN L1(4):=0; END;
IF N>5 THEN L1(4):=2; L1(2):=0; L1(6):=0; END;
ARC_P(93,23,8,L1(1));
ARC_P(123,23,8,L1(2));
ARC_P(153,23,8,L1(3));
ARC_P(123,53,8,L1(4));
ARC_P(93,83,8,L1(5));
ARC_P(123,83,8,L1(6));
ARC_P(153,83,8,L1(7));
FREEZE;
END;
```

Le programme affiche la face du dé tirée aléatoirement.







Algorithme : le pré et la chèvre

HP 39gII

2nde

Une personne possède un pré de forme carrée de 10m de côté. Il attache une chèvre par une corde reliée à un piquet planté au milieu d'un des côtés. Il souhaite que la chèvre broute une surface d'aire égale à la moitié de l'aire du pré.
Quelle longueur de corde doit-il laisser ?

Solution pas à pas :

Le bout de la corde tendue trace un arc de cercle.
Si la longueur de corde est inférieure au côté du pré carré, la surface que peut brouter la chèvre est celle du demi-disque de rayon la longueur de corde.
Si elle est supérieure au côté du carré, la surface est composée d'un rectangle et d'une portion de disque.
Pour connaitre la largeur du rectangle, on utilise l'égalité de Pythagore dans le triangle rectangle ci-contre : $x^2 = 5^2 + \text{Largeur}^2$
Largeur du rectangle = $\sqrt{x^2 - 25}$

Pour calculer l'aire de la portion de disque, on retire à l'aire du secteur angulaire l'aire du triangle rouge :

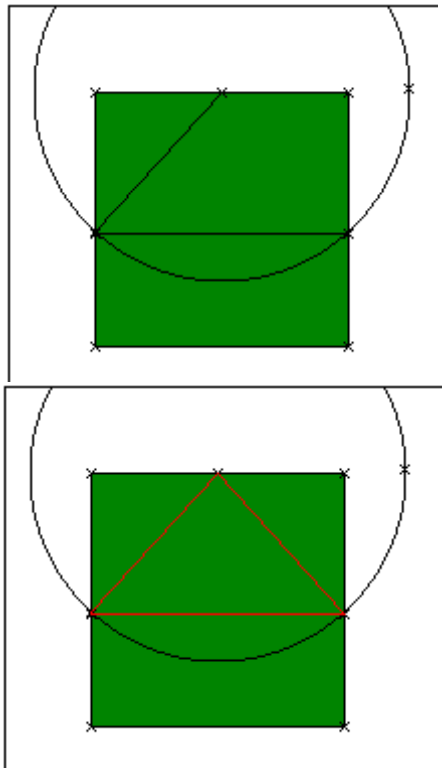
$$\frac{\alpha}{360}\pi x^2 - \frac{10\sqrt{(25 - x^2)}}{2}.$$

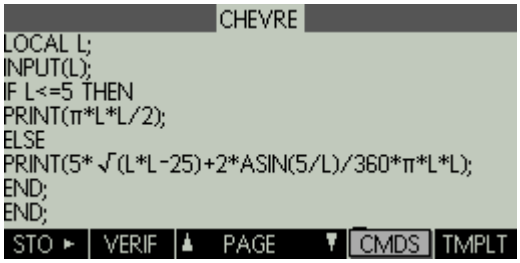
α est l'angle du secteur angulaire qui se calcule par trigonométrie avec un 2.arcsin(5/x).

On peut alors écrire ce programme calculant la surface du pré broutée par la chèvre suivant la longueur de corde saisie :

```
EXPORT CHEVRE()
BEGIN
LOCAL L;
//On demande la longueur de la corde
INPUT(L);
//On traite les 2 cas de surfaces
IF L<=5 THEN
PRINT(π*L*L/2);
ELSE
PRINT(√(L*L−25)*10+2*ASIN(5/L)/360*π*L*L−5*√(L*L−25));
END;
END;
```

Captures d'écran :



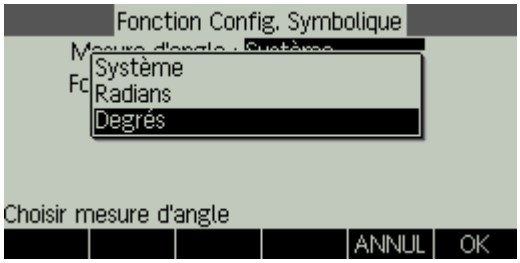


Le pré et la chèvre

HP 39gII

2nde

Attention à bien régler l'unité d'angle en degrés (touches SHIFT Symb Setup).



En exécutant le programme, on trouve une surface de $50m^2 = 100m^2 \div 2$ une longueur de corde d'environ 5,8 m.

Algorithme : vendredi 13

HP 39gII

2nde

Niveau : 2nde

Exercice : Démontrer que tous les ans, il y a au moins un vendredi 13 dans l'année.

Thèmes de programmation : boucles, conditions, utilisation des listes.

Solution pas à pas :

On crée trois listes, une pour les jours (lundi, mardi, etc…), une pour les mois et une contenant le nombre de jours par mois.

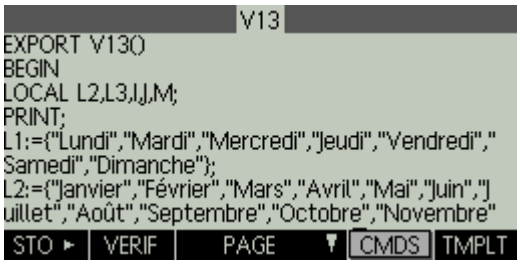
On part ensuite d'une date de départ : le 13 janvier. Selon que cette date tombe un lundi, un mardi, un mercredi, un jeudi, un vendredi, un samedi ou un dimanche, on regarde si un vendredi 13 est atteint en passant en revue chaque mois suivant.

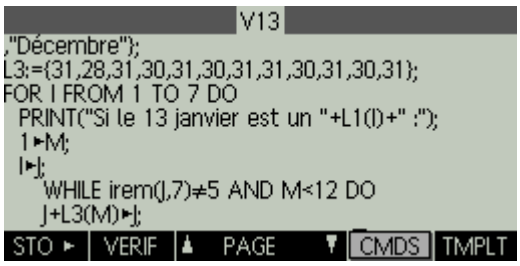
Pour le voir, on ajoute le nombre de jours du mois à la journée de départ et on calcule le reste de cette somme dans une division euclidienne par 7. Si le reste est 5, on tombe sur un vendredi (vendredi est le 5ème jour de la semaine).

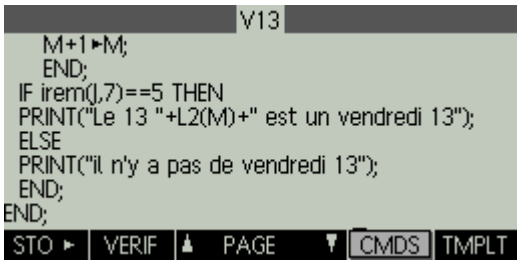
On crée alors ce programme sur HP 39gII :

```
EXPORT V13()
BEGIN
LOCAL L1,L2,L3,I,J,M;
PRINT;
L1:={"Lundi","Mardi","Mercredi","Jeudi","Vendredi","Samedi","D
imanche"};
L2:={"Janvier","Février","Mars","Avril","Mai","Juin","Juillet","Août
","Septembre","Octobre","Novembre","Décembre"};
L3:={31,28,31,30,31,30,31,31,30,31,30,31};
FOR I FROM 1 TO 7 DO
  PRINT("Si le 13 janvier est un "+L1(I)+" :");
  1▸M;
  I▸J;
WHILE irem(J,7)≠5 AND M<12 DO
  J+L3(M)▸J;
     M+1▸M;
     END;
```

Captures d'écran :







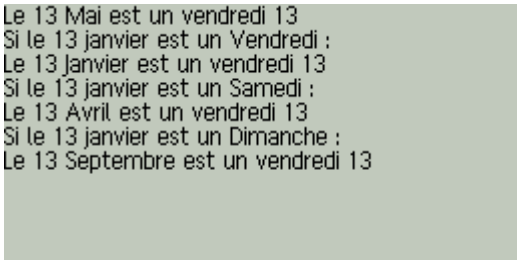
Algorithme : vendredi 13

HP 39gII

2nde

```
  IF irem(J,7)==5 THEN
  PRINT("Le 13 "+L2(M)+" est un vendredi 13");
  ELSE
  PRINT("il n'y a pas de vendredi 13");
  END;
END;
END;
```

L'exécution du programme montre que quelque soit le jour de la semaine du 13 janvier de l'année, on tombe toujours ensuite sur un vendredi 13.



Algorithme : nombre de Kaprekar

HP 39gII

2nde

Un nombre de Kaprekar est un nombre qui, lorsqu'il est élevé au carré, peut être séparé en une partie gauche et une partie droite (non nulle) telles que la somme donne le nombre initial.

Exemple : $4879^2 = 23804641$ et $238 + 04641 = 4879$.

Créer un algorithme vérifiant si un nombre est de Kaprekar.

Thèmes de programmation : boucles, conditions, utilisation des listes.

Solution pas à pas :

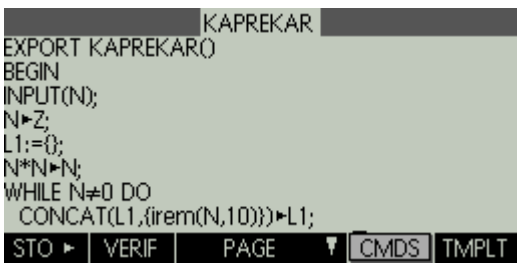
Il faut d'abord extraire chaque chiffre du carré du nombre choisi.
On stocke chacun des chiffres dans une liste.
Pour extraire chaque chiffre, on effectue des divisions euclidiennes successives par 10 et on prend chaque reste.
La commande REVERSE(permet de renverser la liste pour afficher les chiffres tels qu'ils sont écrits de gauche à droite dans le résultat du carré du nombre choisi.

Une fois la liste créée, il faut tester toutes les combinaisons de parties gauche et droite.
Pour les obtenir toutes, on imbrique deux boucles For l'une dans l'autre.
On écrit alors les nombres obtenus avec chaque partie en utilisant des multiplications par 10.
On implante un test d'égalité à la fin de chaque création des deux parties. Si l'égalité de Kaprekar (la somme des deux parties est égale au nombre de départ) est vérifiée, on affiche que le nombre est de Kaprekar (avec éventuellement le détail de la décomposition) sinon, on affiche rien.

On écrira sur HP 39gII :

```
EXPORT KAPREKAR()
BEGIN
INPUT(N);
N▸Z;
L1:={};
N*N▸N;
WHILE N≠0 DO
  CONCAT(L1,{irem(N,10)})▸L1;
  iquo(N,10)▸N;
```

Captures d'écran :

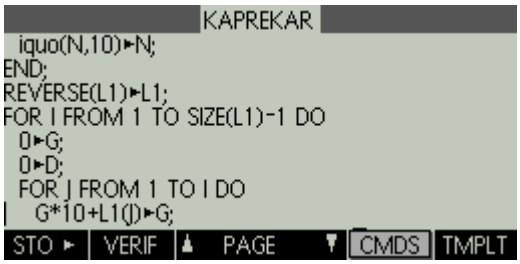


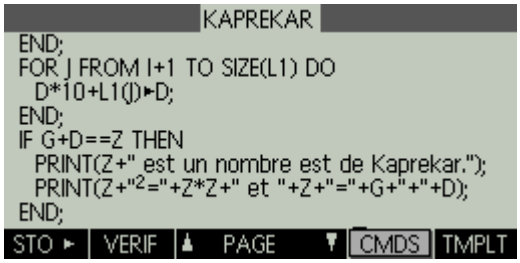
Nombre de Kaprekar

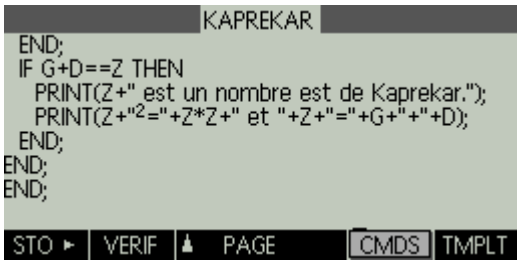
HP 39gII

2nde

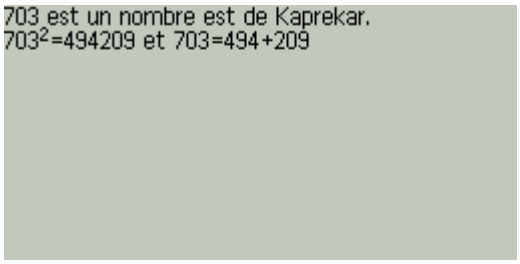
```
END;
REVERSE(L1)▸L1;
FOR I FROM 1 TO SIZE(L1)-1 DO
  0▸G;
  0▸D;
  FOR J FROM 1 TO I DO
    G*10+L1(J)▸G;
  END;
  FOR J FROM I+1 TO SIZE(L1) DO
    D*10+L1(J)▸D;
  END;
  IF G+D==Z THEN
    PRINT(Z+" est un nombre est de Kaprekar.");
    PRINT(Z+"²="+Z*Z+" et "+Z+"="+G+"+"+D);
  END;
END;
END;
```







On peut tester le programme avec par exemple le nombre 703 qui est un nombre de Kaprekar.



Algorithme : dés de Sicherman

HP 39gII

2nde

Sur une idée de Jean-Philippe BLAISE

*Les **dés de Sicherman** sont deux dés à 6 faces : les faces du premier sont consituées des numéros 1, 2, 2, 3, 3 et 4 et celles du second 1, 3, 4, 5, 6 et 8. Lorsque l'on jette ces 2 dés et que l'on ajoute les résultats des faces, on obtient non seulement les mêmes possibilités qu'avec un dé classique (de 2 à 12), mais avec les mêmes fréquences d'apparition !*

Réaliser un programme qui lance cinq cents fois deux dés de Sicherman ainsi que deux dés classiques et compare graphiquement les fréquences des sommes obtenues.

Solution pas à pas :

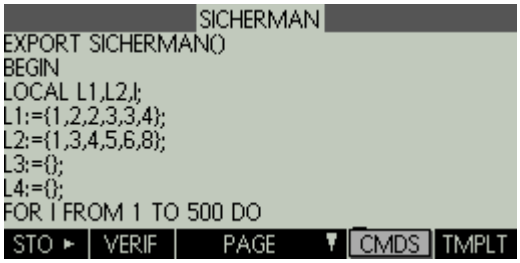
On stocke dans deux listes L3 et L4 les sommes des deux faces obtenues avec les deux types de dés sur 500 lancers (boucle For de 1 à 500).
1+FLOOR(6*RANDOM) donne un nombre entier aléatoire compris entre 1 et 6.

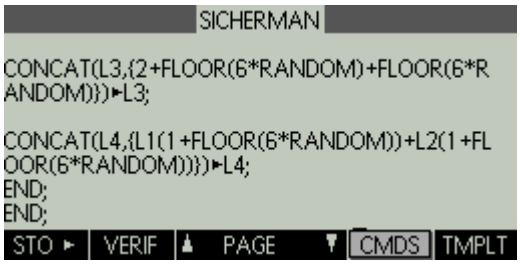
```
EXPORT SICHERMAN()
BEGIN
LOCAL L1,L2,I;
L1:={1,2,2,3,3,4};
L2:={1,3,4,5,6,8};
L3:={};
L4:={};
FOR I FROM 1 TO 500 DO
  CONCAT(L3,{1+FLOOR(6*RANDOM)+
1+FLOOR(6*RANDOM)})▸L3;
  CONCAT(L4,{L1(1+FLOOR(6*RANDOM))+L2(1+
FLOOR(6*RANDOM))})▸L4;
END;
END;
```

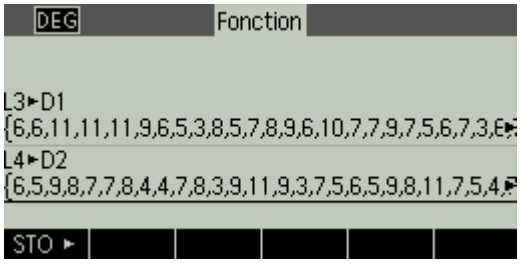
Pour exploiter statistiquement les deux listes créées avec le programme, il faut les stocker dans les variables D1 et D2.

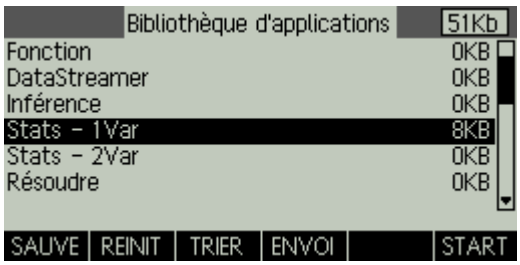
On lance ensuite l'application «Stats – 1Var » accessible depuis la touche Apps Info .

Captures d'écran :







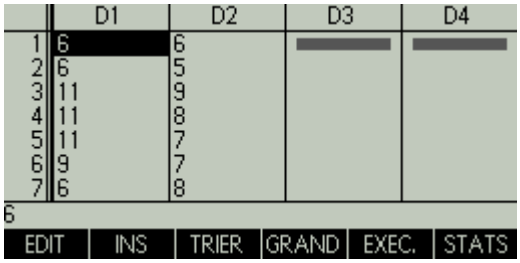


Dés de Sicherman

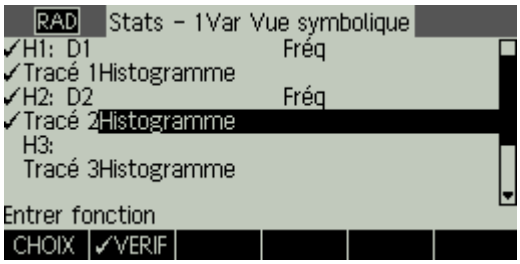
HP 39gII

2nde

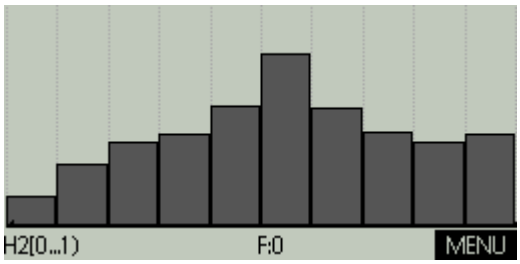
Les deux listes créées avec le programme apparaissent dans les 2 premières colonnes.



On appuie sur la touche Symb Setup pour effectuer les réglages sur le diagramme à afficher.
On sélectionne le tracé en histogramme et on saisit D2 au niveau de H2.
On cochera dans un premier temps D1 qui affichera l'histogramme obtenu avec les dés normaux.



Une pression sur la touche Plot Setup donne cet histogramme.



Appuyer à nouveau sur la touche Symb Setup pour cette fois cocher H2.

Une pression sur la touche Plot Setup donne l'histogramme pour les lancers des deux dés de Sicherman.
L'histogramme présente la même forme que celui des dés normaux.

Plus on augmente le nombre de lancers, plus l'histogramme de Sicherman s'approche de celui des deux dés normaux.

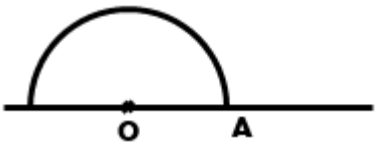
Algorithme : tracer une spirale

HP 39gII

2nde

Niveau : Seconde.

Enoncé : Tracer une spirale obtenue en traçant des demi-cercles centrés successivement en O et en A.

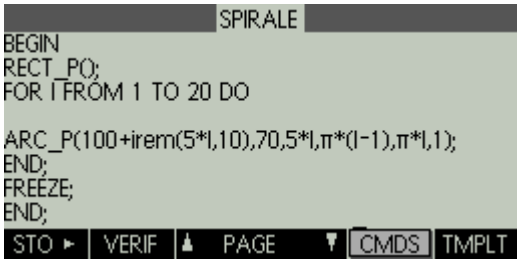


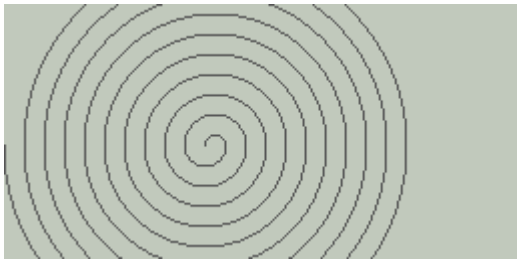
Solution pas à pas :

On réalise 20 demi-cercles en partant d'un demi-cercle de rayon 5.
La HP 39gII trace des arcs de cercle avec la commande *ARC_P(x,y,R,a1,a2,C)*
où (x,y) sont les coordonnées du centre, R le rayon, a1 et a2 précisent l'angle délimité par l'arc et C sa couleur.
Pour faire varier successivement les centres des demi-cercles du point O au point A, on additionne à l'abscisse d'origine le reste des rayons successifs dans leurs divisions euclidiennes par le double du rayon. On additionne ainsi 0 ou le rayon successivement.
Les demi-cercles se tracent successivement avec des écarts d'angles entre 0 et π puis entre π et 2π. On peut alors utiliser dans la boucle incrémenté sur I les valeurs $(I-1)\pi$ et $I\pi$.
RECT_P() ; permet d'obtenir un écran vierge avant le tracé.
FREEZE ; permet d'arrêter l'écran sur le dessin.

```
EXPORT SPIRALE()
BEGIN
RECT_P();
FOR I FROM 1 TO 20 DO
  ARC_P(100+irem(5*I,10),70,5*I,π*(I−1),π*I,1);
END;
FREEZE;
END;
```

Captures d'écran :





Algorithme : suite de Syracuse

HP 39gII

2nde

La suite de Syracuse est définie par $u_0 \in \mathbb{N}^*$ et

$$\text{Pour tout } n \in \mathbb{N},\ u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

1. Calculer les premiers termes de la suite pour $u_0 = 1$; $u_0 = 3$ et $u_0 = 7$.

On ne sait pas à l'heure actuelle s'il existe un entier u_0 pour lequel cette suite n'atteint jamais 1.

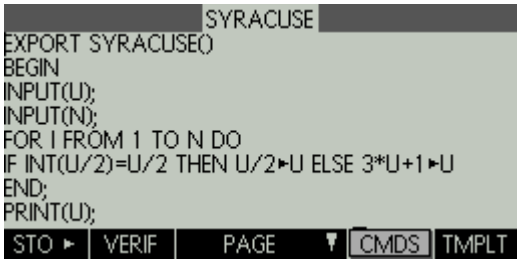
2. Ecrire un programme demandant u_0 et n à l'utilisateur et affichant toutes les valeurs $u_1 ; u_2 ; \ldots ; u_n$.

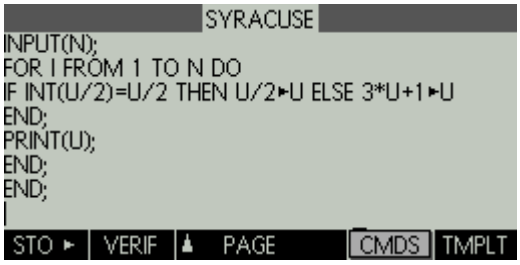
Solution pas à pas :

Algorithme
Entrée
Demander à l'utilisateur le terme initial u_0
et un entier n
Initialisation
Stocker la valeur u_0 dans u
Traitement
Pour i allant de 1 à n
Si u est pair (si $u/2$ est entier) alors
Stocker $u/2$ dans u
Afficher u
Sinon
Stocker $3*u+1$ dans u
Afficher u
Fin du si
Fin de la boucle pour
Sortie
Afficher p

Captures d'écran :

Sur HP-39gII, on écrira :





Algorithme : numéro de SIRET

HP 39gII

2nde

Chaque entreprise possède un code unique l'identifiant : le numéro de SIRET (Système d'Identification du Répertoire des Etablissements).
Le code SIRET comporte 14 chiffres, le dernier étant une clé de contrôle.
Il est composé de cette manière :
On positionne chaque chiffre du code du rang 14 au rang 1.
On multiplie les chiffres de rang impair par 1 et ceux de rang pair par 2.
On additionne les chiffres de chaque résultat de multiplication.
On additionne les résultats de chaque rang.
Si la somme est un multiple de 10, le numéro SIRET est valide.
Exemple : SIRET du ministère de l'Education Nationale : 11004301500012

14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	**1**	**0**	**0**	**4**	**3**	**0**	**1**	**5**	**0**	**0**	**0**	**1**	**2**
1x2	1x1	0x2	0x1	4x2	3x1	0x2	1x1	5x2	0x1	0x2	0x1	1x2	2x1
2	1	0	0	8	3	0	1	(10) 1+0 =1	0	0	0	2	2

2+1+0+0+8+3+0+1+1+0+0+0+2+2=20 qui est un multiple de 10.

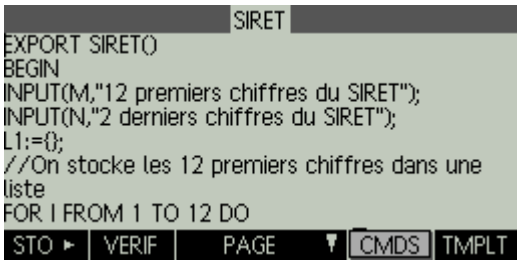
Créer un algorithme vérifiant un numéro SIRET.

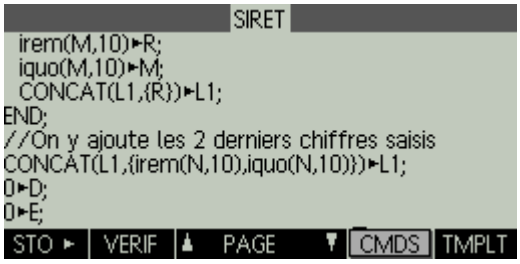
Solution pas à pas :

On demande à l'utilisateur de saisir un numéro SIRET.
La HP 39gII gère les nombres jusqu'à 12 chiffres. Il faut donc séparer la demande à l'utilisateur en deux : les 12 premiers chiffres puis les 2 derniers.
Voici le programme avec annotations explicatives :

```
EXPORT SIRET()
BEGIN
INPUT(M,"12 premiers chiffres du SIRET");
INPUT(N,"2 derniers chiffres du SIRET");
L1:={};
//On stocke les 12 premiers chiffres dans une liste
FOR I FROM 1 TO 12 DO
  irem(M,10)▶R;
  iquo(M,10)▶M;
  CONCAT(L1,{R})▶L1;
END;
//On y ajoute les 2 derniers chiffres saisis
CONCAT(L1,{irem(N,10),iquo(N,10)})▶L1;
0▶D;
0▶E;
//On double tous les chiffres de rang pair
FOR I FROM 1 TO 7 DO
  L1(2*I)*2▶P;
```

Captures d'écran :





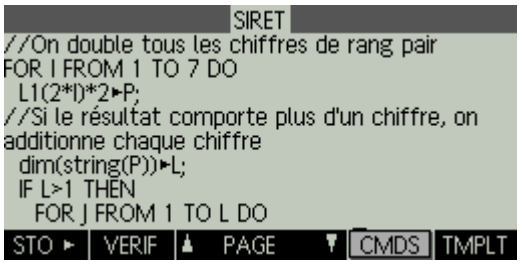
Numéro de SIRET

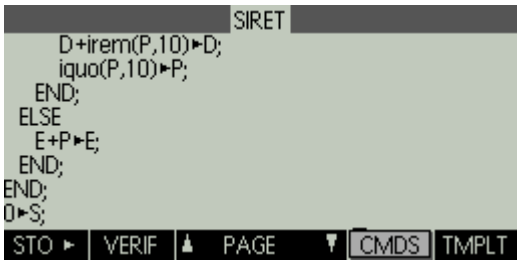
HP 39gII

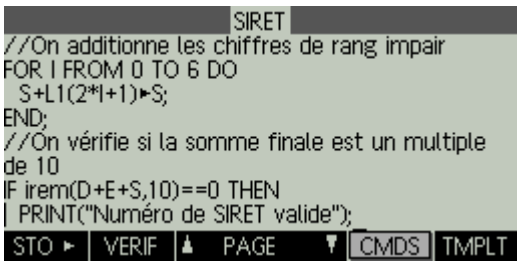
2nde

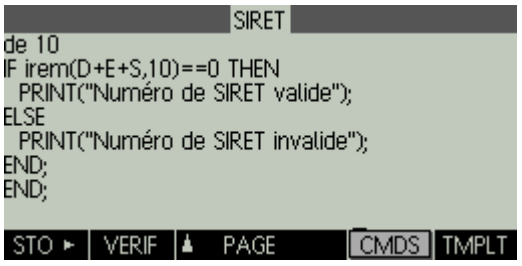
```
//Si le résultat comporte plus d'un chiffre, on additionne chaque chiffre
  dim(string(P))▸L;
  IF L>1 THEN
    FOR J FROM 1 TO L DO
      D+irem(P,10)▸D;
      iquo(P,10)▸P;
    END;
  ELSE
    E+P▸E;
  END;
END;
0▸S;
//On additionne les chiffres de rang impair
FOR I FROM 0 TO 6 DO
  S+L1(2*I+1)▸S;
END;
//On vérifie si la somme finale est un multiple de 10
  IF irem(D+E+S,10)==0 THEN
    PRINT("Numéro de SIRET valide");
  ELSE
    PRINT("Numéro de SIRET invalide");
  END;
END;
```

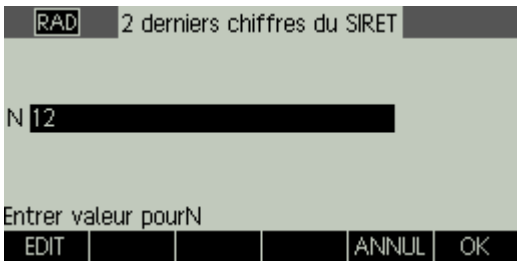
On saisit le numéro SIRET (en deux fois : 12 chiffres puis les 2 derniers) et le programme affiche si le numéro est valide ou invalide.

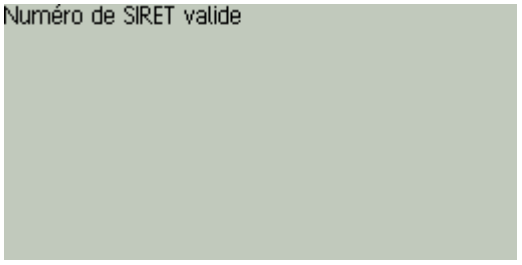












Algorithme : numéro ISBN

HP 39gII

2nde

Chaque livre publié est identifié par un code unique : le numéro ISBN (International Standard Book Number).
Le code ISBN comporte 10 chiffres, le dernier étant une clé de contrôle.
Le code est validé ainsi : on additionne les neufs premiers chiffres multipliés chacun par leur rang. Le reste du résultat de cette addition dans la division euclidienne par 11 doit être la clé (le dernier chiffre).
Remarque : si le reste est 10, le dernier chiffre est noté X.
Exemple : avec l'ISBN 2501086902 (mini guide des champignons).

1	2	3	4	5	6	7	8	9	10
2	**5**	**0**	**1**	**0**	**8**	**6**	**9**	**0**	**2**
2x1	5x2	0x3	1x4	0x5	8x6	6x7	9x8	0x9	
2	10	0	4	0	48	42	72	0	

2+10+0+4+0+48+42+72+0=178=11x16+2 et 2 est bien le dernier chiffre.

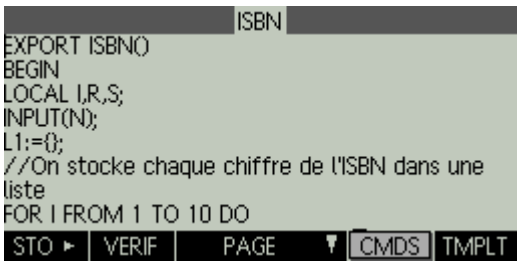
Solution pas à pas :

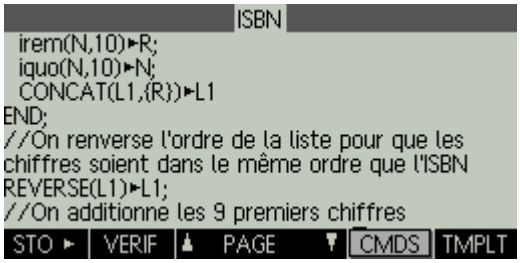
On demande à l'utilisateur de saisir un numéro ISBN (10 chiffres).

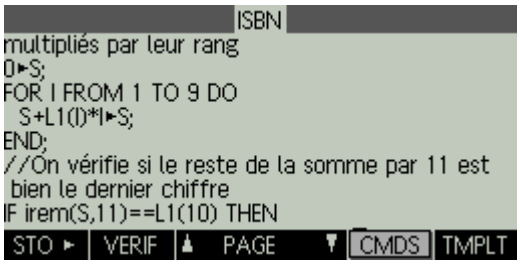
Voici le programme avec annotations explicatives :

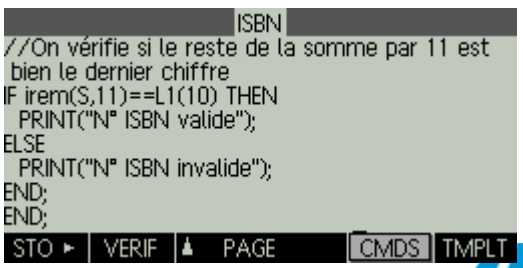
```
EXPORT ISBN()
BEGIN
LOCAL I,R,S;
INPUT(N);
L1:={};
//On stocke chaque chiffre de l'ISBN dans une liste
FOR I FROM 1 TO 10 DO
  irem(N,10)▸R;
  iquo(N,10)▸N;
  CONCAT(L1,{R})▸L1
END;
//On renverse l'ordre de la liste pour que les chiffres soient dans le même ordre que l'ISBN
REVERSE(L1)▸L1;
//On additionne les 9 premiers chiffres multipliés par leur rang
0▸S;
FOR I FROM 1 TO 9 DO
  S+L1(I)*I▸S;
END;
//On vérifie si le reste de la somme par 11 est bien le dernier chiffre
IF irem(S,11)==L1(10) THEN
  PRINT("N° ISBN valide");
ELSE
  PRINT("N° ISBN invalide");
END;
END;
```

Captures d'écran :









Algorithme : jeu des allumettes

HP 39gII

2nde

Ce jeu se joue à 2 joueurs.
On commence avec 10 allumettes. A tour de rôle, chaque joueur enlève entre 1 et 3 allumettes.
Celui qui retire la dernière allumette a perdu la partie.

Créer un programme permettant de jouer à ce jeu.

Solution pas à pas :

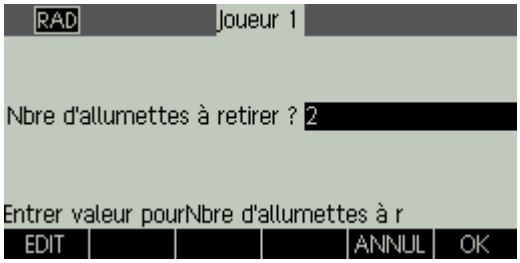
Voici le programme avec annotations explicatives :

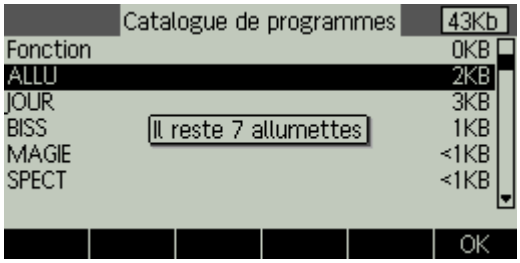
```
EXPORT ALLU()
BEGIN
LOCAL N,J,M,X,Y,I;
//On établit au départ le nombre d'allumettes à 10 et on règle le 1er joueur sur 1
10▸N;
1▸J;
//Tant qu'il reste plus d'une allumette, on enchaîne les coups en alternant les joueurs
WHILE N>1 DO
 INPUT(M,"Joueur "+J,"Nbre d'allumettes à retirer ?");
 IF M>3 THEN
  MSGBOX("3 allumettes maximum !") ;
 ELSE
  IF J==1 THEN 2▸J; ELSE 1▸J; END;
  N-M▸N;
 END;
 MSGBOX("Il reste "+N+" allumettes");
END;
//On indique quel joueur a perdu
MSGBOX("Le joueur "+J+" a perdu !");
END;
```

Bonus : on peut améliorer le programme en créant une interface graphique :

```
EXPORT ALLU()
BEGIN
LOCAL N,J,M,I;
//On établit au départ le nombre d'allumettes à 10 et on règle le 1er joueur sur 1
10▸N
1▸J;

//On affiche 10 rectangles pour représenter les allumettes
```

Captures d'écran :

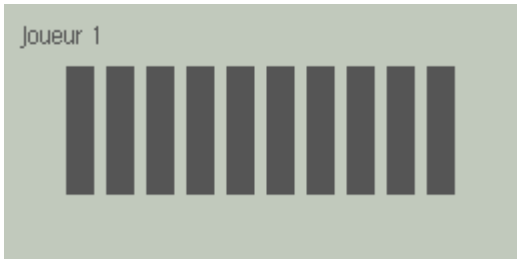


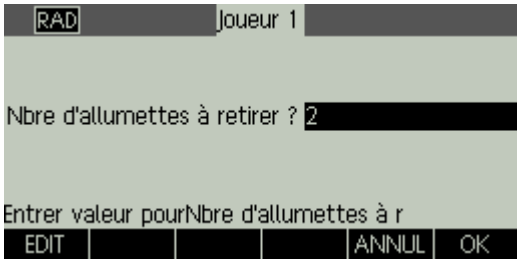


Jeu des allumettes

HP 39gII

```
//On dessine 10 rectangles pour représenter les allumettes
RECT_P;
TEXTOUT_P("Joueur "+J,10,10,1,1) ;
FOR I FROM 1 TO 10 DO
RECT_P(10+20*I,30,25+20*I,62,3,1);
END;
//On affiche les allumettes pendant 5 secondes
WAIT(5);
//Tant qu'il reste plus d'une allumette, on enchaîne les coups
en alternant les joueurs
WHILE N>1 DO
 INPUT(M,"Joueur "+J,"Nbre d'allumettes à retirer ?");
 IF M>3 THEN
  MSGBOX("3 allumettes maximum !") ;
 ELSE
  IF J==1 THEN 2▸J; ELSE 1▸J; END;
  N-M▸N;
 END;
//On affiche les allumettes restantes
RECT_P;
TEXTOUT_P("Joueur "+J,10,10,1,1);
FOR I FROM 1 TO N DO
 RECT_P(10+20*I,30,25+20*I,95,3,1);
END;
WAIT(5);
END;
//On indique quel jour a perdu
MSGBOX("Le joueur "+J+" a perdu !");
END;
```





Algorithme : chronomètre

HP 39gII

Programmer un chronomètre sur la HP 39gII.

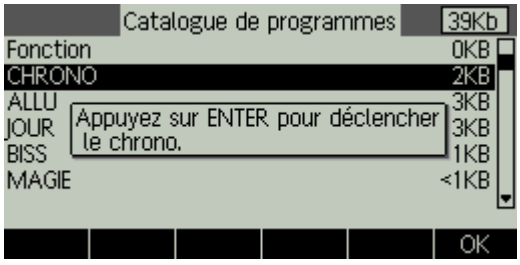
Solution pas à pas :

On exploite des commandes très utiles de la HP 39gII :
la commande WAIT(1) permettant d'attendre 1 seconde et la commande ISKEYDOWN pour déclencher et arrêter le chronomètre par pression d'une touche.

Voici le programme avec annotations explicatives :

```
EXPORT CHRONO()
BEGIN
LOCAL S,M;
//On initialise les secondes et les minutes à 0
0▸S;
0▸M;
//On prépare le lancement du chrono
MSGBOX("Appuyez sur ENTER pour déclencher le chrono");
//On affiche le chrono et gérant l'incrémentation des minutes
toutes les 60 secondes
REPEAT;
  WAIT(1);
  S+1▸S;
  IF S==60 THEN M+1▸M; 0▸S; END;
  RECT_P;
  TEXTOUT_P(M+":"+S,10,10,2,1);
UNTIL ISKEYDOWN(50)==1
FREEZE;
END;
```

Captures d'écran :



Algorithme : problème du spaghetti

HP 39gII

Problème du Spaghetti

Je dispose d'un spaghetti. Quelle est la probabilité qu'en le coupant en trois je puisse former avec les trois bouts obtenus un triangle ?

Solution pas à pas :

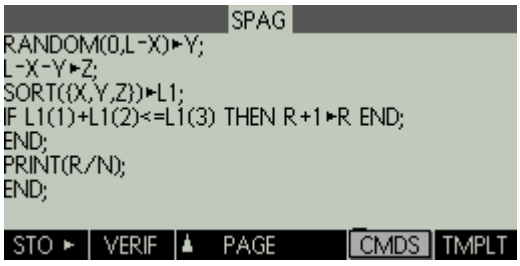
Il s'agit de vérifier l'inégalité triangulaire sur les trois longueurs de spaghetti obtenu aléatoirement.
On fixera la longueur totale du spaghetti.
On peut ainsi établir l'algorithme suivant :

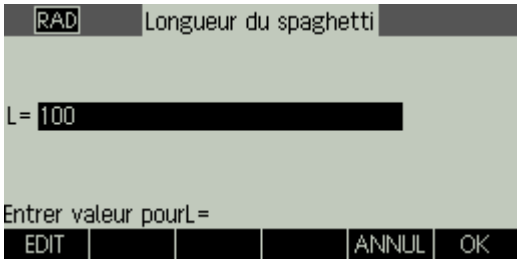
Algorithme
Entrée
Demander le nombre d'essais N
Demander la longueur du spaghetti L
Initialisation
Initialisation de la variable R (nombre de succès)
Traitement
Pour I allant de 1 à N
Couper le 1er morceau de longueur X
(X = aléatoire tel que 0 < X < L)
Couper le 2nd morceau de longueur Y
(Y = aléatoire tel que 0 < Y < L–X)
Calculer la longueur du 3ème morceau Z
(Z= L–X–Y)
Si le maximum de ces trois longueurs est inférieur ou égal à la somme des deux autres
Alors Augmenter R de 1
Fin du Si
Fin du Pour
Sortie
Afficher R/N

L'algorithme retourne la fréquence de triplets vérifiant l'inégalité triangulaire.
Plus le nombre d'essais est grand, plus la fréquence tend vers la probabilité recherchée.

Captures d'écran :

Sur HP-39gII, on écrira :



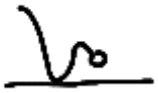


Algorithme : balle rebondissante

HP 39gII

2nde

On lance une balle d'une hauteur initiale de 300 cm.
On suppose qu'à chaque rebond, la balle perd 10 % de sa hauteur (la hauteur est donc multipliée par 0,9 à chaque rebond).
On cherche à savoir le nombre de rebonds nécessaire pour que la hauteur de la balle soit inférieure ou égale à 10 cm.
Écrire un algorithme permettant de résoudre ce problème.



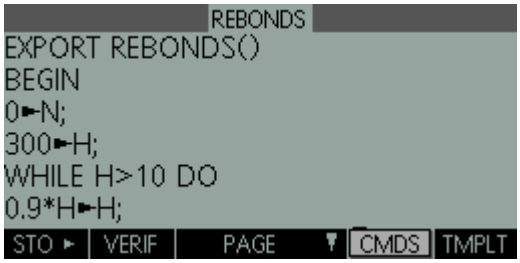
Solution pas à pas :

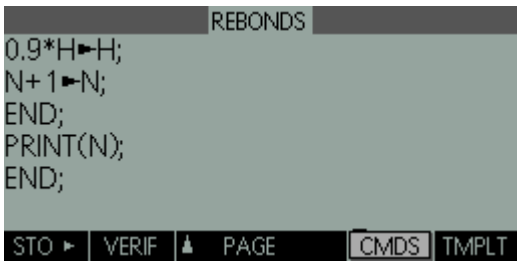
On réduit successivement la hauteur précédente de 10% depuis la hauteur initiale jusqu'à ce que les 10 cm soient atteints.
On utilisera une boucle « Tant que » dans l'algorithme :

Algorithme
Initialisation
Nombre *h* initialisé à la valeur 300
Nombre *n* initialisé à la valeur 0
Traitement
Tant que *h* > 10
Stocker *h**0,9 dans *h*
Stocker *n*+1 dans *n*
Fin de la boucle tant que
Sortie
Afficher *n*

Captures d'écran :

Sur HP-39gII, on écrira :





Algorithme : limitation des naissances

HP 39gII

2nde

Niveau : Seconde.
Objectifs : vérifier une conjecture, écrire et utiliser un algorithme.
Mots-clés : probabilités, algorithme, itération, boucle While.

Enoncé : Dans un pays, on limite le nombre de naissances de filles en :

• ayant au maximum 4 enfants pour chaque famille ;
• arrêtant les naissances dès la naissance d'un garçon.

Quelle conséquence sur la population a cette politique de natalité ?

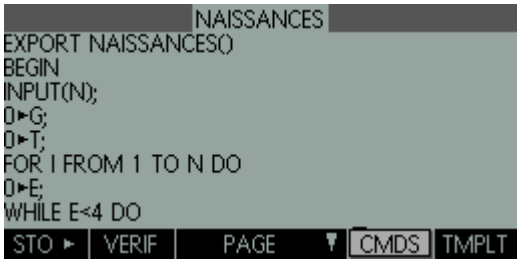
Solution pas à pas :

On réalise une simulation avec l'algorithme suivant affichant la fréquence d'apparition d'un garçon :

Variables :
N : nombre de familles
G : nombre de garçons au total
F : nombre de filles d'une famille
E : nombre d'enfants d'une famille
T : nombre d'enfants nés au total
Traitement :
Saisir N
Initialiser G à 0
Initialiser T à 0
Pour I variant de 1 à N
Initialiser E à 0
Initialiser F à 0
Tant que E<4 faire
Tirer au hasard entier S entre 1 et 2
E prend la valeur E+1
T prend la valeur T+1
Si S=1
Alors G prend la valeur G+1
Sinon F prend la valeur F+1
Fin Si
Fin Tant que
Fin Pour
Sortie :
Afficher G/T
Fin

La création du programme sur HP 39gII depuis les touches [SHIFT] [1 Prgm X] donne : *recopier le code des images ci-contre.*

Captures d'écran :



Limitation des naissances

HP 39gII

2nde

```
NAISSANCES
WHILE E<4 DO
ROUND(1+RANDOM,0)▸S;
E+1▸E;
T+1▸T;
IF S=1 THEN
G+1▸G;
END;
END;
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

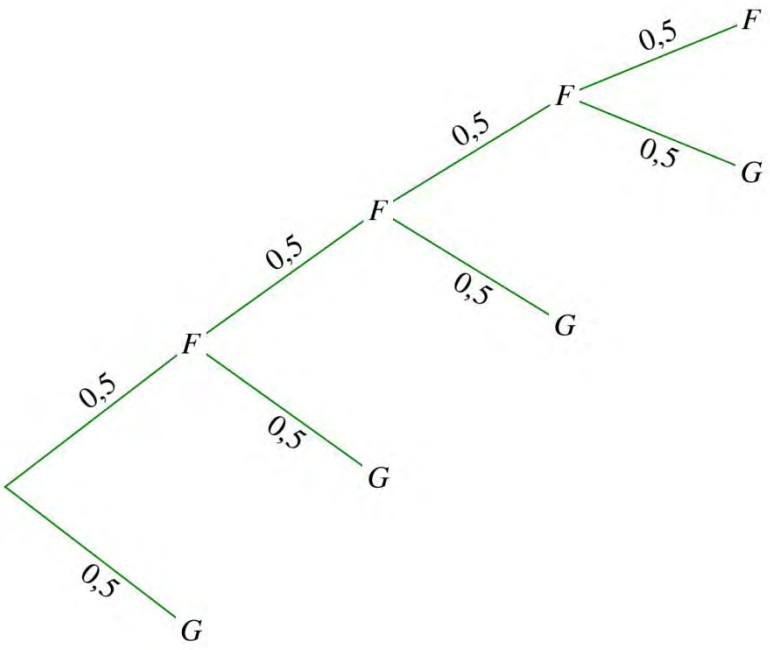
```
NAISSANCES
IF S=1 THEN
G+1▸G;
END;
END;
END;
PRINT(G/T);
END;
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

```
RAD  Entrée

N 500

Entrer valeur pourN
EDIT | | | | ANNUL | OK
```

```
.4995
```

Si on lance l'algorithme pour un grand nombre de familles, la fréquence des garçons est très proche de 0,5. Cette politique de natalité ne semble donc n'avoir aucune conséquence sur le nombre de garçons.

On peut en effet démontrer que cela ne change rien en dressant un arbre de probabilités et en calculant les espérances :



Limitation des naissances

HP 39gII

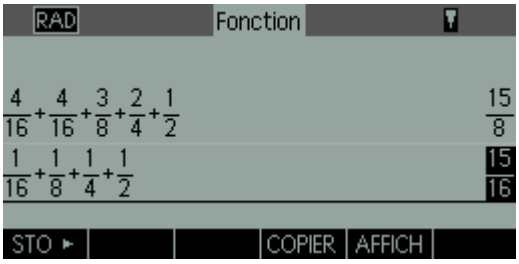
2nde

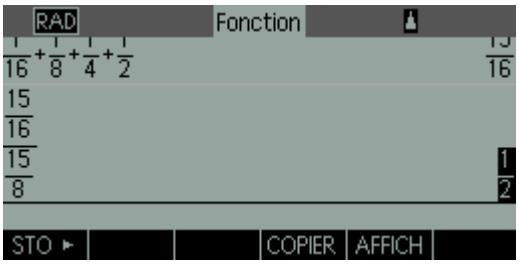
On peut alors résumer les résultats dans ce tableau :

Nombre N d'enfants	Nombre G de garçons	Probabilité
4	0	1/16
4	1	1/16
3	1	1/8
2	1	1/4
1	1	1/2

On en déduit alors les espérances :
E(N) = 4 × 1/16 + 4 × 1/16 + 3 × 1/8 + 2 × 1/4 + 1 × 1/2 = 15/8
E(G) = 1 × 1/16 + 1 × 1/8 + 1 × 1/4 + 1 × 1/2 = 15/16
et E(G)/E(N) = 1/2.

On appuie sur la touche $\frac{a\leftrightarrow b/c}{\angle\leftrightarrow\circ\prime\prime\ C}$ pour obtenir une valeur exacte en écriture fractionnaire.





Régression linéaire

HP 39gII

2nde

En statistiques à 2 variables, une régression linéaire (ou encore ajustement affine) est une méthode utilisée pour modéliser l'une des variables en fonction de l'autre et ainsi faire des estimations ou des prédictions.
On recherche une relation de type affine entre les deux variables qui graphiquement est représentée par une droite passant le plus près possible de tous les points.
La HP 39gII utilise la méthode des moindres carrés pour obtenir cette droite.

Solution pas à pas :

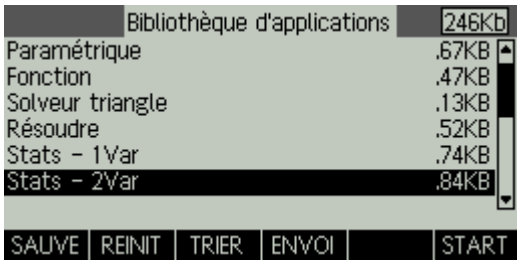
Lancer l'applet Statistiques à 2 vars (touche Apps Info).

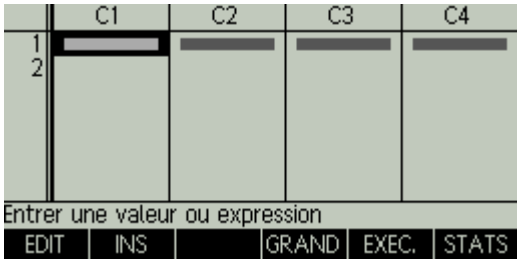
Entrer dans la colonne C1 la première liste de valeurs puis dans C2 la seconde liste.

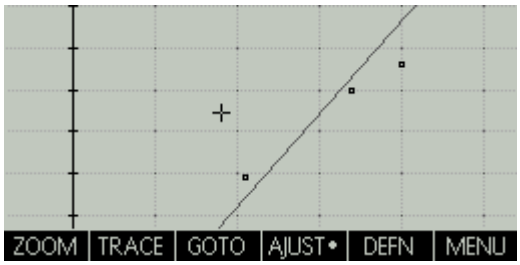
Appuyer ensuite sur Plot Setup et l'ajustement affine se fait directement !

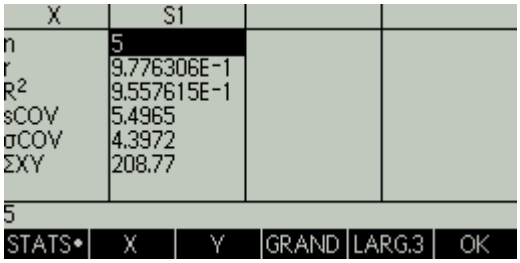
Appuyer sur la touche Num Setup et appuyer sur F6 (STATS) pour obtenir un résumé statistique.

Captures d'écran :









Etude de suite récurrente

HP 39gII

Tle ES

Exercice type : On considère la suite (u_n) définie par $u_0 = 1$ et, pour tout entier n,

$$u_{n+1} = \sqrt{2u_n}.$$

1/ On considère l'algorithme suivant :

Variables :	n est un entier naturel
	u est un réel positif
Initialisation :	Demander la valeur de n
	Affecter à u la valeur 1
Traitement :	Pour i variant de 1 à n :
	| Affecter à u la valeur $\sqrt{2u}$
	Fin de Pour
Sortie :	Afficher u

a/ Donner une valeur approchée à 10^{-4} près du résultat qu'affiche cet algorithme lorsqu'on choisit $n = 3$.

b/ Que permet de calculer cet algorithme ?

2/ Recopier l'algorithme ci-dessous et le compléter par les instructions du traitement et de la sortie, de façon à afficher en sortie la plus petite valeur n telle que $u_n > 1{,}999$.

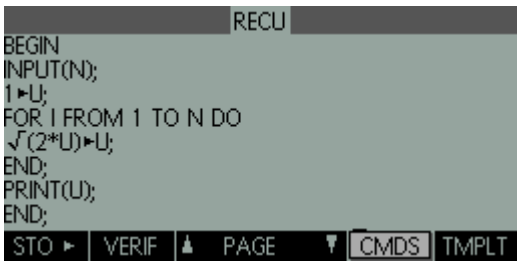
Variables :	n est un entier naturel
	u est un réel
Initialisation :	Affecter à n la valeur 0
	Affecter à u la valeur 1
Traitement :	
Sortie :	

Solution pas à pas :

1/ a/ On programme l'algorithme sur la HP 39gII depuis l'éditeur de programmes (touches SHIFT 1 Prgm X).

On écrit le programme ci-contre :

Captures d'écran :

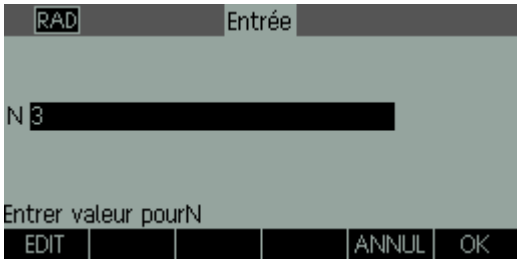


Etude de suite récurrente

HP 39gII

Tle ES

On fait tourner l'algorithme pour N=3, on obtient une valeur approchée de U_3.

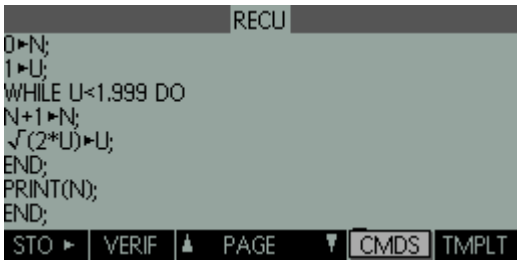


b/ L'algorithme calcule le Nième terme de la suite (Un).

Sur HP-39gII, on écrira :

2/ Il faut calculer les termes successifs de la suite récurrente jusqu'on dépasse 1,999. On utilisera donc une boucle « Tant que » :

Variables :
N : entier naturel
U : réel
Traitement :
Initialiser N à 0
Initialiser U à 1
Tant que U<1,999 faire
N prend la valeur N+1
U prend la valeur $\sqrt{U+1}$
Fin Tant que
Sortie :
Afficher N
Fin



On fait tourner l'algorithme pour obtenir la valeur *n* recherchée.

Simulation d'un lancer de dé

HP 39gII

2nde

Simuler le lancer d'un dé à 6 faces sur la HP 39gII.

Solution pas à pas :

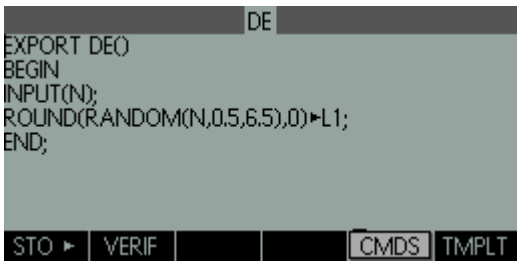
L'expérience aléatoire consiste à tirer au hasard un nombre entier compris entre 1 et 6.

La HP 39gII dispose d'une commande RANDOM qui retourne un nombre aléatoire compris entre deux bornes.

On crée un programme simulant N lancers et stockant les résutalts dans une liste.
Pour générer un nombre entier aléatoire entre 1 et 6, on prend l'arrondi à l'unité d'un nombre choisi aléatoirement sur l'intervalle [0,5 ; 6,5] avec la commande RANDOM.
Le programme stocke la liste dans L1.

Appuyer sur les touches SHIFT 7 (List P) pour accéder à L1.

Captures d'écran :



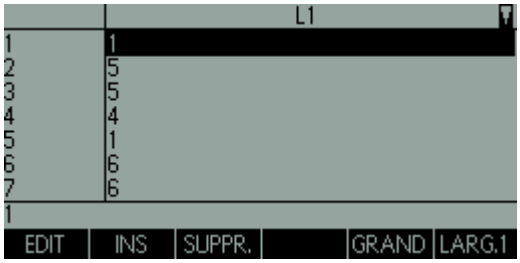


Schéma de Bernoulli : loi binomiale

HP 39gII

Exercice type : Une urne contient 49 boules blanches et une boule dorée. On gagne quand on tire la boule dorée.

1/ Calculer la probabilité de tirer une boule blanche et celle de gagner.
2/ Montrer qu'il s'agit d'une épreuve de Bernouilli en précisant les paramètres.
3/ On effectue 5 fois un tirage avec remise. Calculer les probabilités de gagner 0 fois, 1 fois, 2 fois, 3 fois , 4 fois et 5 fois.
4/ Représenter ces probabilités par un diagramme en bâtons.

Solution pas à pas :

1/ p(« tirer une boule blanche ») = 49/50 = 0,98.
p(« tirer une boule dorée ») = 1 – 0,98 = 0,02.

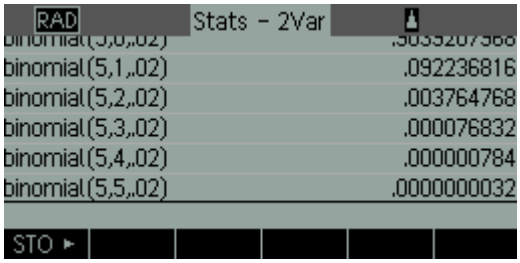
2/ L'expérience est à 2 issues possibles : tirer une boule blanche où l'on perd et tirer une boule dorée où l'on gagne. On est donc dans un schéma de Bernoulli de paramètres n = nombre de tirages et p = probabilité de gagner = 0,02.

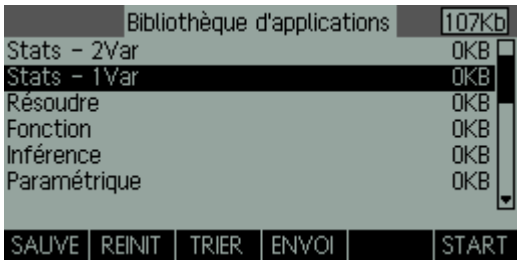
3/ La HP 39gII dispose de la commande **binomial(n,k,p)** qui calcule la probabilité de gagner k fois sur un schéma de Bernoulli de paramètres (n,p).
Ici n = 5 tirages.
On obtient les probabilités recherchées avec cette commande.

4/ On lance l'application « Stats – 1 Var » depuis la touche Apps Info .

On entre dans la colonne D1 les 6 valeurs de probabilités calculées précédemment.

Captures d'écran :





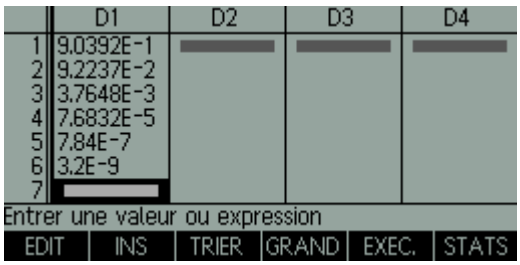
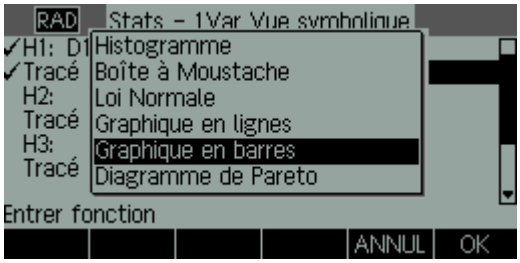


Schéma de Bernoulli : loi binomiale

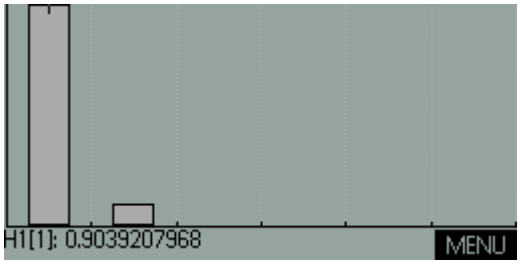
HP 39gII

Appuyer sur la touche Symb Setup pour choisir le type de diagramme.



Appuyer sur la touche Views Help pour sélectionner l'échelle automatique.

Seules 2 barres sont visibles, la hauteur des 4 autres étant très proches de 0 (probabilités très faibles).



Optimisation à 2 variables : régionnement du plan

Divers

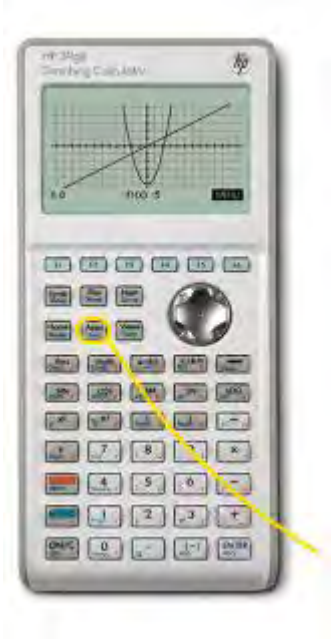
HP 39gII

Niveau : Terminale STG (ancien programme)

Exercice type BAC : Un artisan potier et ses ouvriers fabriquent deux modèles différents de poteries :

- Le modèle A qui nécessite 500 g de terre et 4 heures de travail.
- Le modèle B qui nécessite 800 g de terre et 3 heures de travail.

Par semaine, l'artisan commande 30 kg de terre et son entreprise fournit au plus 170 heures de travail.
Le four ne peut pas cuire plus de 45 poteries par semaine.
On note x le nombre de poteries du modèle A et y le nombre de poteries du modèle B.
L'artisan réalise un bénéfice de 6€ par poterie de modèle A et 5€ par poterie de modèle B.
On admet que toute la production hebdomadaire est vendue.
Il cherche à déterminer le nombre de poteries de chaque modèle qu'il doit fabriquer chaque semaine pour réaliser un bénéfice maximal.



1/ Traduire par une inéquation les contraintes sur la quantité de terre.
2/ Traduire par une inéquation les contraintes sur les heures de travail.
3/ Traduire par une inéquation la contrainte liée à la cuisson dans le four.
4/ Traduire l'ensemble des contraintes par un système.
5/ Résoudre graphiquement ce système.

Touche d'accès aux fonctions :

Solution pas à pas :

1/ La quantité de terre en kg utilisée pour la poterie A est 0,5 x.
La quantité de terre en kg utilisée pour la poterie A est 0,8 y.
On ne peut utiliser que 30 kg de terre maximum.
Donc $\mathbf{0{,}5\,x + 0{,}8\,y \leq 30}$.

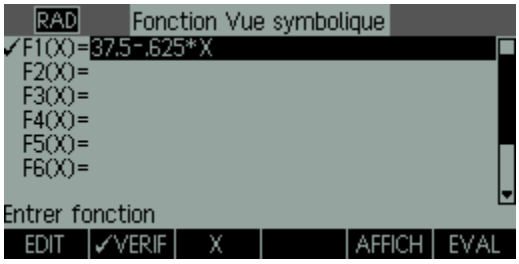
Accéder aux applications depuis la touche Apps Info .
Aller sur Fonction avec les touches fléchées et appuyer sur F6 .

Captures d'écran :

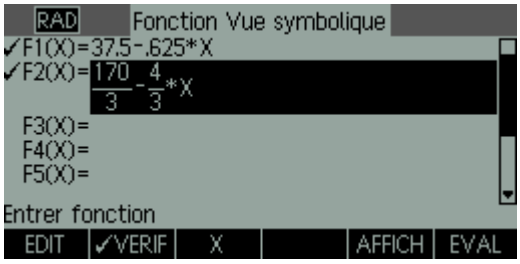
Optimisation à 2 variables : régionnement du plan

HP 39gII

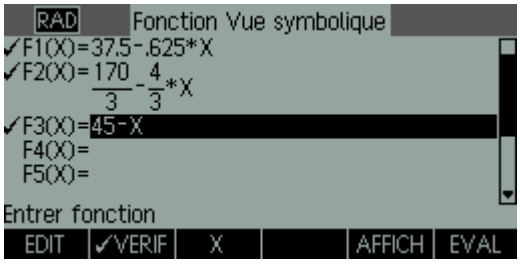
Il faut isoler y :
$y \leq$ **(30 – 0,5 x)÷0,8**.
C'est-à-dire : $y \leq$ **37.5 – 0,625 x**
On rentre alors la fonction.



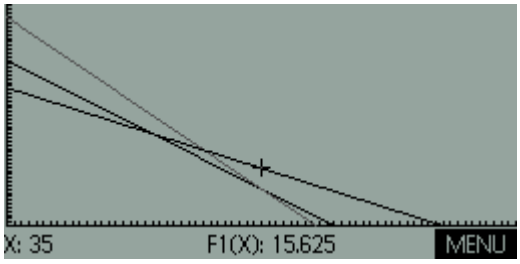
2/ Le temps de travail utilisé pour la poterie A est $4x$.
Le temps de travail utilisé pour la poterie A est $3y$.
Il n'y a que 170 heures de travail maximum.
Donc **$4x + 3y \leq 170$**.
C'est-à-dire : $y \leq$ **170/3 – 4/3 x**
On rentre alors la fonction.



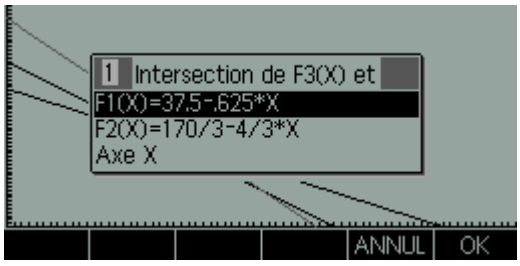
3/ On fabrique $x + y$ poteries par semaine.
On ne peut cuire que 45 poteries maximum par semaine.
Donc **$x + y \leq 45$**.
C'est-à-dire : $y \leq$ **45 – x**
On rentre alors la fonction.



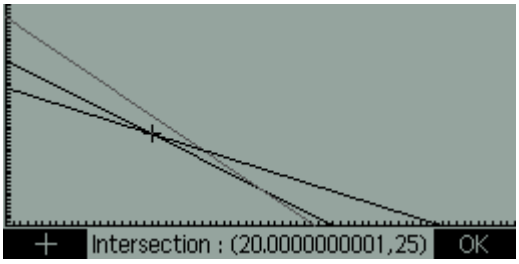
4/ En appuyant sur la touche Plot Setup, on obtient les trois droites.
a zone solution se situe sous les droites.
On désire maximiser le bénéfice **$B = 6x + 5y$**
Ce qui donne les droites parallèles de bénéfices :
$y =$ **B/5 – 6/5 x**
La plus haute d'entre-elles coupant la zone des solutions est celle passant par l'intersection de la droite représentative de F1(X) et celle de F3(X).



Pour trouver l'intersection, placer le curseur sur la droite de F3(X), appuyer sur F6 (MENU) puis F4.
Choisir Intersection… et sélectionner F1(X).



La HP 39gII place automatiquement le curseur à l'intersection et donne ses coordonnées :
Le bénéfice est maximal pour :
x = 20 poteries de modèle A et
y = 25 poteries de modèle B.



Position relative de deux droites

HP 39gII

Ecrire un programme qui, avec les équations de deux droites, indique leur position.

Solution pas à pas :

On demandera à l'utilisateur de spécifier les 6 coefficients des deux équations de droites :
(d_1) : Ax+By+C=0 (avec A≠0 ou B≠0)
(d_2) : Dx+Ey+F=0 (avec D≠0 ou E≠0)

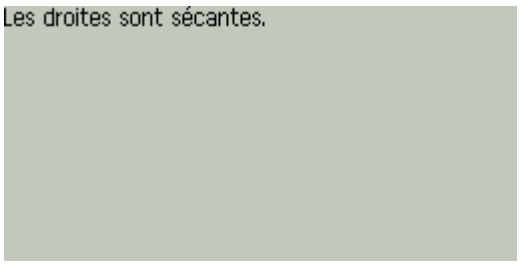
On vérifie ensuite le parallélisme des droites en vérifiant la colinéarité des vecteurs directeurs : on vérifie si A*E=D*B.

On traite le cas de droites confondues : C*E=D*B

Cela donne sur HP 39gII :

```
EXPORT DROITES()
BEGIN
LOCAL A,B,C,D,E,F;
//On demande à l'utilisateur les 6 coefficients
INPUT(A,"dans Ax+By+C=0 avec A≠0 ou B≠0");
INPUT(B,"dans Ax+By+C=0 avec A≠0 ou B≠0");
INPUT(C,"dans Ax+By+C=0");
INPUT(D,"dans Cx+Dy+E=0 avec C≠0 ou D≠0");
INPUT(E,"dans Cx+Dy+E=0 avec C≠0 ou D≠0");
INPUT(F,"dans Cx+Dy+E=0");
//On vérifie la colinéarité des vecteurs directeurs
IF A*E<>D*B THEN
  PRINT("Les droites sont sécantes.");
ELSE
  IF C*E==D*B THEN
    PRINT("Les droites sont confondues.");
  ELSE
    PRINT("Les droites sont strictement parallèles.");
  END;
END;
END
```

Captures d'écran :



Etude de trinôme du second degré

HP 39gII

1ère S

Etude des zéros et des variations des fonctions trinomiales du second degré :

$X \rightarrow AX^2 + BX + C$

Solution pas à pas :

Accéder à l'application « Résoudre » depuis la touche [Apps Info] et la lancer ([F6]).

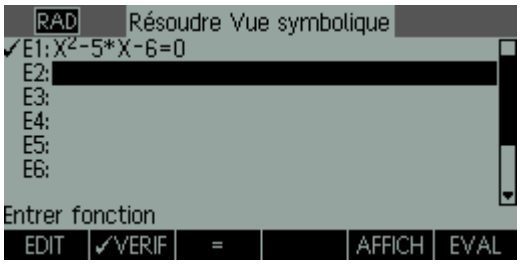
On s'intéressera ici à l'équation :
$X^2 - 5X - 6 = 0$.

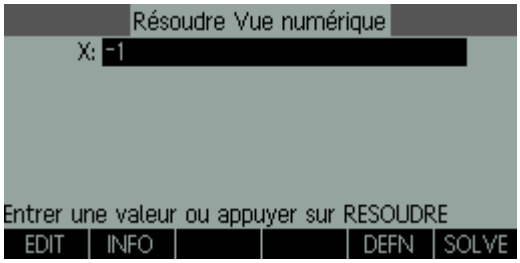
Appuyer sur la touche [Num Setup] puis sur F6 (SOLVE).
la HP 39gII retourne alors une solution : x = -1.

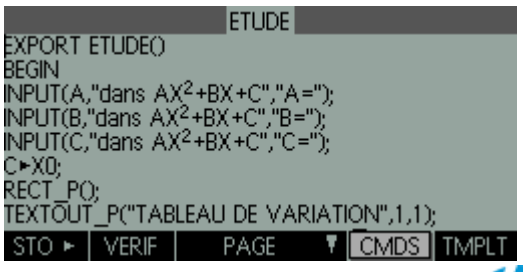
Sélectionner ALT dans le menu pour obtenir l'autre solution X = 6.

Les capacités de programmation de la HP 39gII sont suffisamment puissantes pour établir directement sur la calculatrice le tableau de variation d'un trinôme du second degré.
On écrira le programme ci-contre sur la HP 39gII :

Captures d'écran :







Etude de trinôme du second degré

HP 39gII

1ère S

```
ETUDE
TEXTOUT_P("f(X)="+A+"X²+"+B+"X+"+C,1,15);
RECT_P(1,30,250,100,1);
TEXTOUT_P("-∞",3,30,0,3);
TEXTOUT_P("+∞",230,30,0,3);
LINE_P(1,45,250,45,3);
IF A≠0 THEN
-B/(2*A)▸X0;
A*X0^2+B*X0+C▸E;
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

```
ETUDE
ROUND(E,2)▸E;
IF 2*A*(X0-0.1)+B<0 THEN
LINE_P(10,50,120,90,3);
LINE_P(130,90,240,50,3);
TEXTOUT_P(E,110,75,0,3);
ELSE
LINE_P(10,90,120,50,3);
LINE_P(130,50,240,90,3);
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

```
ETUDE
TEXTOUT_P(E,110,60,0,3);
END;
ELSE
IF B==0 THEN
TEXTOUT_P("Fonction constante",80,50,0,3);
END;
IF B<0 THEN
LINE_P(10,50,240,90,3);
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

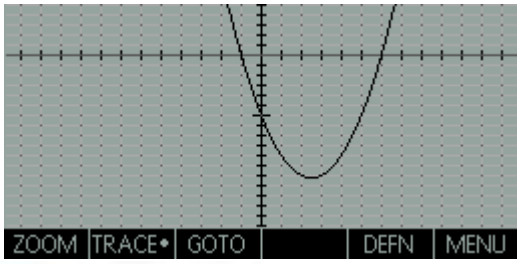
```
ETUDE
ELSE IF B>0 THEN
LINE_P(10,90,240,50,3);
END;
END;
END;
TEXTOUT_P(X0,120,30,0,3);
FREEZE();
END;
STO ▸ | VERIF | PAGE | CMDS | TMPLT
```

On lance le programme qui invite à rentrer les valeurs des coefficients A, B et C.

```
RAD  dans AX²+BX+C

A= 1

Entrer valeur pourA=
EDIT | | | | ANNUL | OK
```

Le programme retourne alors le tableau de variation du trinôme.
On peut vérifier en jetant un coup d'œil sur la représentation graphique du trinôme depuis la touche Plot Setup .

```
TABLEAU DE VARIATION
f(X)=1X²+-5X+-6
-∞        2.5        +∞
          -12.25
```



Algorithme : racines réelles d'un trinôme

HP 39gII

1ère S

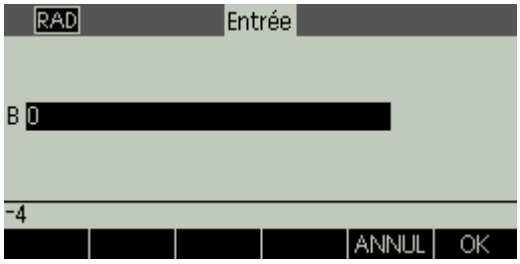
Programmer un algorithme donnant les racines réelles d'un trinôme du second degré ax^2+bx+c

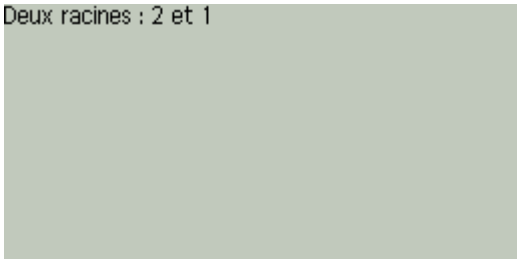
Solution pas à pas :

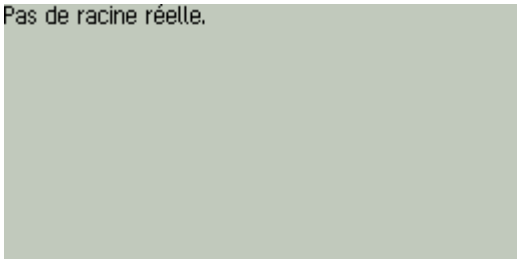
On créer un programme TRINOME depuis l'éditeur (touches SHIFT 1 Prgm X) et on tape l'algorithme suivant :

```
EXPORT TRINOME()
BEGIN
LOCAL A,B,C,D;
//On demande à l'utilisateur de saisir les 3 coefficients du trinôme
INPUT(A);
INPUT(B);
INPUT(C);
//On traite le cas où A=0
IF A==0 THEN
  PRINT("Racine ="+(-C/B));
ELSE
//Pour les autres cas, on calcule le discriminant
  B*B-4*A*C▶D;
//On donne les racines suivant la valeur du discriminant
  IF D==0 THEN
    PRINT("Racine ="+(-B/(2*A)) ;
  END;
  IF D>0 THEN
    PRINT("Deux racines : "+((-B-√(D))/(2*A))+ " et "+((-B+√(D))/(2*A)));
  END ;
  IF D<0 THEN
    PRINT("Il n'y a pas de racine réelle.");
  END;
END;
END;
```

Captures d'écran :







Algorithme : triangle de Pascal

HP 39gII

Construire par algorithme le tableau ci-dessous.
La première colonne est composée de 1 et chaque autre valeur du tableau est obtenue en additionnant la case du dessus et la voisine de gauche de cette dernière.

n= \ p=	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

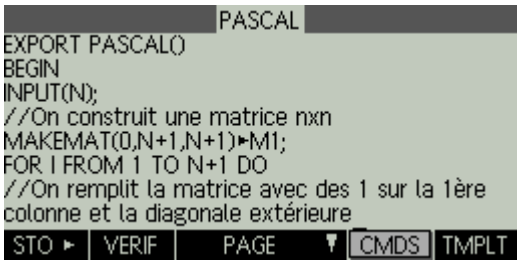
Solution pas à pas :

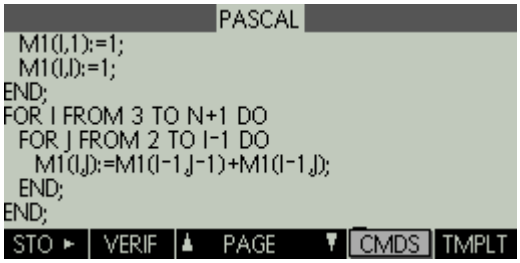
On demande à l'utilisateur d'entrer la taille du triangle souhaité (valeur de n).
Pour créer le triangle de Pascal, il est intéressant et facile d'utiliser une matrice. On construit donc une matrice nxn et on définit chaque coefficient à l'aide de la formule d'addition expliquée.

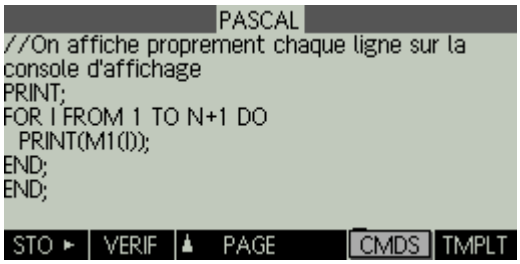
```
EXPORT PASCAL()
BEGIN
INPUT(N);
//On construit une matrice nxn
MAKEMAT(0,N+1,N+1)▸M1;
FOR I FROM 1 TO N+1 DO
//On remplit la matrice avec des 1 sur la 1ère colonne et la diagonale extérieure
  M1(I,1):=1;
  M1(I,I):=1;
END;
FOR I FROM 3 TO N+1 DO
  FOR J FROM 2 TO I-1 DO
    M1(I,J):=M1(I-1,J-1)+M1(I-1,J);
  END;
END;
//On affiche proprement chaque ligne sur la console d'affichage
PRINT;
FOR I FROM 1 TO N+1 DO
  PRINT(M1(I));
END;
END;
```

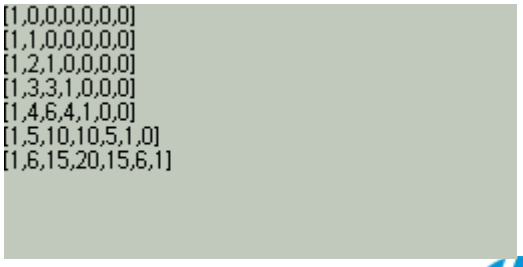
Voici ce que l'on obtient pour n=6.

Captures d'écran :









1ère S

Triangle de Pascal

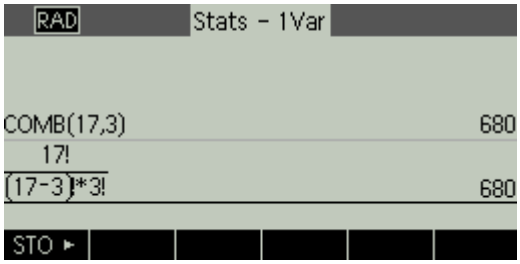
HP 39gII

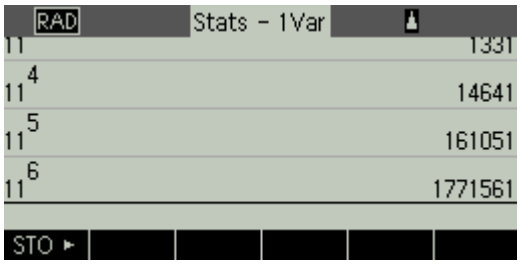
Le nombre situé à l'intersection de la ligne n et de la colonne p représente le coefficient de rang p dans le développement de $(x+y)^n$ (formule du binôme de Newton).
Ce nombre est appelé coefficient binomial et est noté C(n,p).
Il s'exprime par la formule :

$$C(n,p) = \frac{n!}{(n-p)!\times p!}$$

La HP 39gII dispose de la commande COMB(calculant directement ces coefficients binomiaux.

Terminons sur une petite astuce : pour obtenir rapidement une ligne du triangle de Pascal, on peut faire un usage ingénieux de la formule du binôme de Newton : on élève à la puissance le rang de la ligne 11 (sur 4 lignes) puis 101 (sur 4 lignes) puis 1001 (sur 4 lignes), etc…





Test de primalité de Lucas Lehmer

HP 39gII

Soit n un entier impair. n est premier si et seulement s'il existe un entier a tel que a(n-1) = 1 (modulo n) et, pour tout diviseur premier q de (n-1), $a^{(n-1)/q} \neq 1$ (modulo n).

Le test de primalité de Lucas-Lehmer pour les nombres de Mersenne s'applique de la façon suivante :

Soit $M_p = 2^p - 1$ le nombre de Mersenne à tester. On définit une suite de la façon suivante : $s_0=4$; et $s_i = s_{i-1}^2-2$

Le nombre de Mersenne M_p est premier si et seulement si $s_{p-2} = 0$ (modulo M_p).

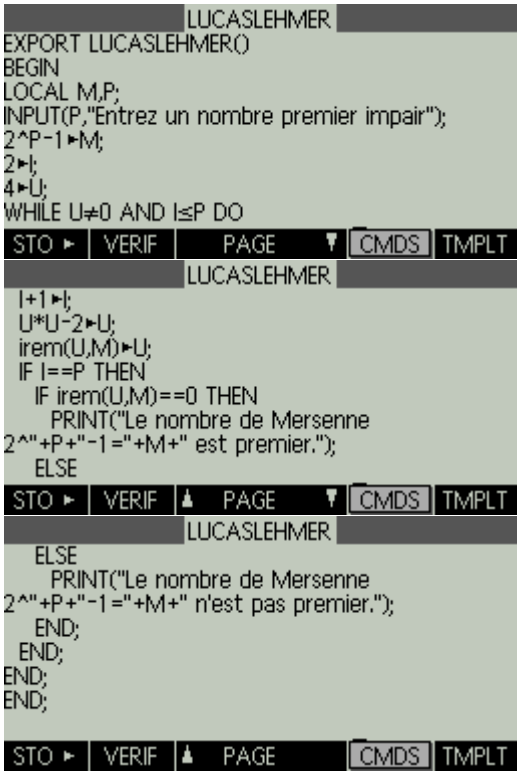
Ecrire un algorithme testant avec cette méthode la primalité d'un nombre de Mersenne.

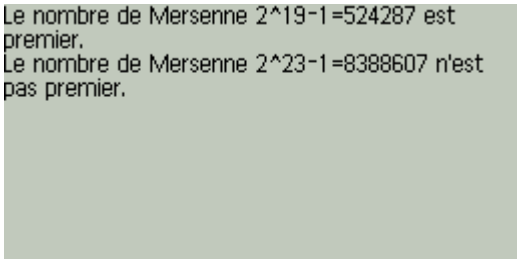
Solution pas à pas :

On fait calculer à l'algorithme les termes successifs de la suite et on effectue un test sur la condition nécessaire et suffisante arrivé au rang souhaité.

```
EXPORT LUCASLEHMER()
BEGIN
LOCAL M,P;
INPUT(P,"Entrez un nombre premier impair");
2^P-1▸M;
2▸I;
4▸U;
WHILE U≠0 AND I≤P DO
 I+1▸I;
 U*U-2▸U;
 irem(U,M)▸U;
 IF I==P THEN
  IF irem(U,M)==0 THEN
   PRINT("Le nombre de Mersenne 2^"+P+"-1="+M+" est
premier.");
  ELSE
   PRINT("Le nombre de Mersenne 2^"+P+"-1="+M+" n'est
pas premier.");
  END;
 END;
END;
END;
```

Captures d'écran :





Tangente à une courbe

HP 39gII

Exercice type : Donner l'équation de la tangente à la courbe représentative de fonction $x \rightarrow e^{-2x-1}$ en 30.
Tracer la tangente.

Solution pas à pas :

Accéder aux applications depuis la touche [Apps Info].
Aller sur Fonction avec les touches fléchées et appuyer sur [F6].

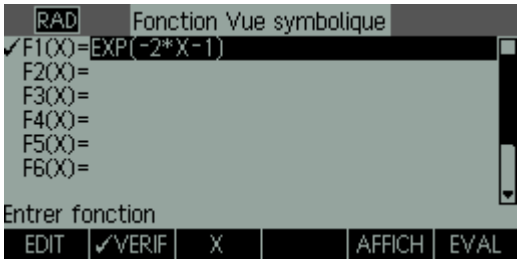
Entrer l'expression algébrique de la fonction à côté de F1(X)=

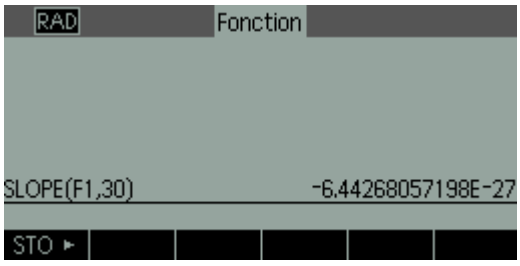
L'équation de la tangente à une courbe représentation d'une fonction f en x = a est donnée par :
y = f '(a)(x – a) + f(a).
La HP 39gII dispose d'une commande calculant la dérivée d'une fonction en une valeur donnée :
SLOPE(F1,30) donne la valeur de la dérivée de notre fonction en 30.

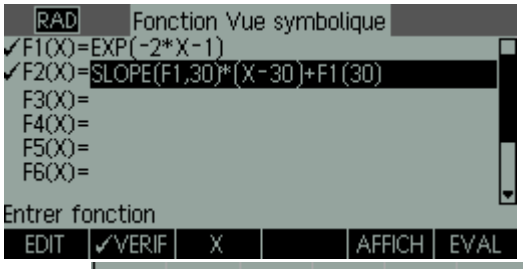
Pour obtenir directement le tracé de la tangente en 0 de la fonction F1, on peut taper à côté de F2(X)=

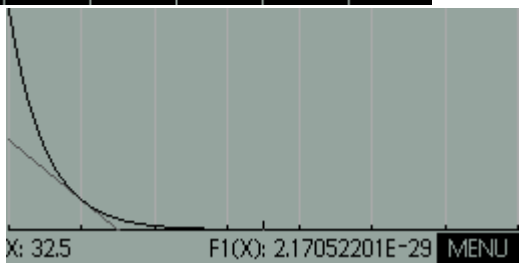
La touche [Plot Setup] donne les tracés.

Captures d'écran :









Encadrement d'une intégrale

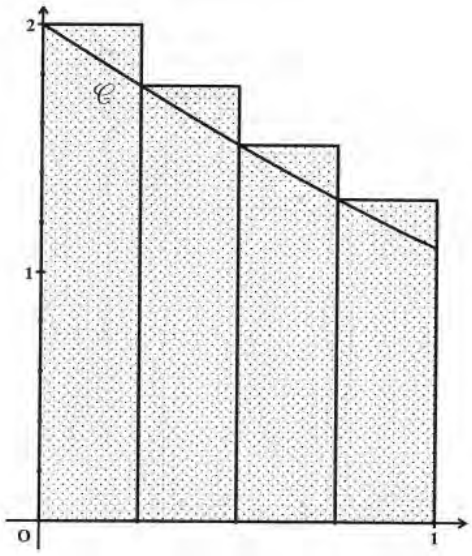
HP 39gII

Ts

Niveaux : Terminales S.
Objectifs : vérifier une conjecture, écrire et utiliser un algorithme.
Mots-clés : algorithme, intégrale, aire.

Enoncé : On considère la fonction f définie sur **R** par $f(x) = (x + 2)e^{-x}$.
On note $\mathcal{C}$ la courbe représentative de la fonction f dans un repère orthogonal.

1/ Etudier les variations de la fonction f sur **R**.
2/ On note $\mathcal{D}$ le domaine compris entre l'axe des abscisses, la courbe $\mathcal{C}$ et les droites d'équation
$x = 0$ et $x = 1$.
On approche l'aire du domaine $\mathcal{D}$ en calculant une somme d'aires de rectangles.
On découpe l'intervalle [0 ; 1] en quatre intervalles de même longueur.



Créer un algorithme permettant d'obtenir une valeur approchée de l'aire du domaine $\mathcal{D}$ en ajoutant les aires des quatre rectangles précédents.
3/ Donner une valeur approchée à 10^{-3} près de l'aire obtenu avec cet algorithme.
4/ On découpe maintenant l'intervalle [0 ; 1] en N intervalles identiques.
Modifier l'algorithme afin qu'il affiche en sortie la somme des aires des N rectangles identiques.

Encadrement d'une intégrale

HP 39gII

Solution pas à pas :

1/ L'étude du signe de la dérivée de la fonction donne ses variations : une croissance sur]-∞ ; -1] et une décroissance sur [-1 ; +∞ [. La fonction est décroissante sur notre intervalle d'étude [0 ; 1].
On peut entrer la fonction dans la HP 39gII et avoir un aperçu de son allure.

2/ Dans le programme, on crée une boucle FOR cumulant les aires des rectangles.
L'aire d'un rectangle s'obtient en multipliant sa largeur 1/4 (1 divisé par le nombre de rectangles) et sa longueur : f (0) pour le premier rectangle, f ((*k*–1)/4) pour le *k*-ième rectangle.

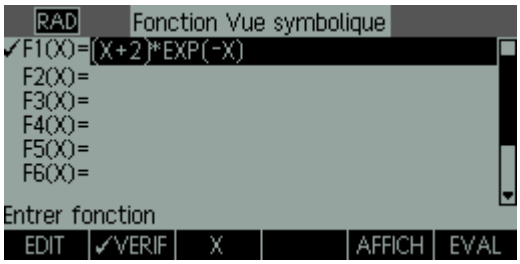
A et B désignent les bornes de l'intervalle d'étude.

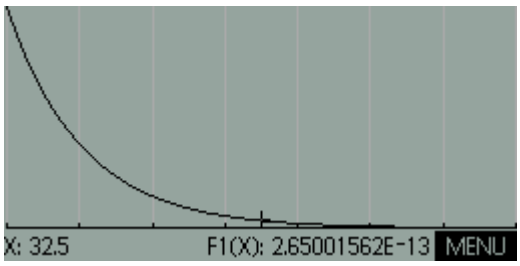
On peut enrichir le programme en faisant même tracer les rectangles avec la commande RECT_P

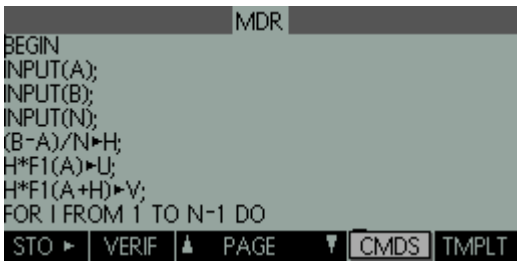
3/ On fait tourner l'algorithme qui donne une valeur approchée par excès puis une par défaut de l'aire sous la courbe :

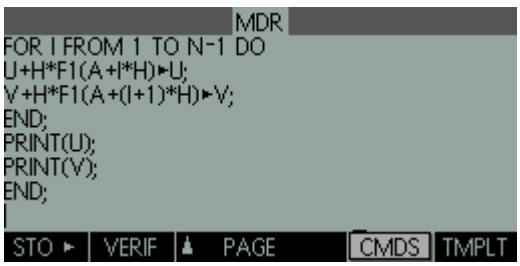
4/ Il suffit juste d'ajouter un INPUT(N); en début de programme pour demander à l'utilisateur le nombre de rectangles dans le découpage et de remplacer les 4 rectangles par N rectangles.

Captures d'écran :









Calcul d'aire entre deux courbes

HP 39gII

TS

D'après une épreuve pratique du BAC S, juin 2008.

Niveau : Terminale S.

Objectifs : fonction, interprétation géométrique d'une intégrale d'une différence entre deux fonctions.

Mots-clés : fonctions, intégrales, aire.

Enoncé : Trouver l'aire entre la courbe représentative de la fonction $f(x) = \ln(x)$ et celle représentative de la fonction $g(x) = (\ln(x))^2$ pour x variant entre 1 et e.

Solution pas à pas :

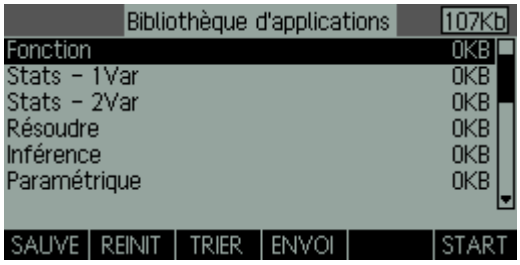
Voici une recherche de solution à ce problème avec la calculatrice graphique HP 39gII.
Accéder d'abord à l'application *Fonction* depuis la touche [Apps Info] .

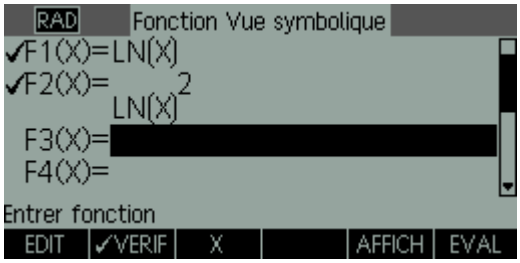
Entrer les deux fonctions f et g à côté de F1(X)= et F2(X)=.

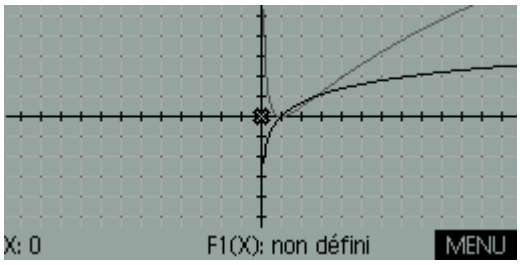
Appuyer sur la touche [Plot Setup] pour voir les courbes représentatives (qui s'affichent en deux niveaux de gris différents pour une belle visibilité).

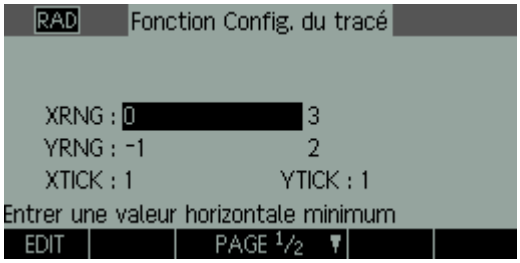
On peut régler les bornes et l'échelle du graphique depuis la combinaison [SHIFT] [Plot Setup] .
Les deux fonctions étant définies pour $x > 0$, on réglera l'abscisse minimum à 0.
L'énoncé demande un calcul d'aire pour x variant de 1 à e, on réglera l'abscisse maximale à 3.
On règle l'ordonnée minimale à -1 et la maximale à 2.

Captures d'écran :





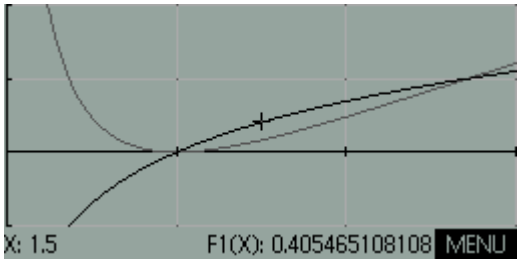




Calcul d'aire entre deux courbes

HP 39gII

Appuyer à nouveau sur la touche [Plot Setup] pour visualiser les courbes et l'aire de la surface les séparant sur l'intervalle demandé.

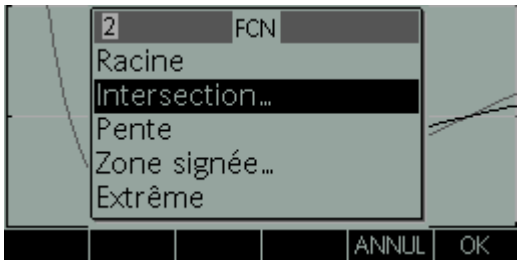


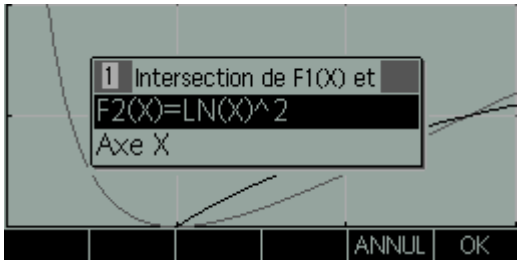
On comprend l'intervalle d'étude puisque les deux courbes se coupent en $x = 1$ et $x = e$.

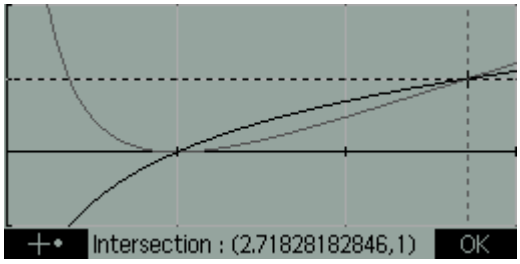
Appuyer sur [F6] (MENU) pour activer les outils d'analyse.

La HP 39gII renvoie les coordonnées des points d'intersection de deux courbes depuis la touche [F4].

Choisir « Intersection… » puis F2(X).

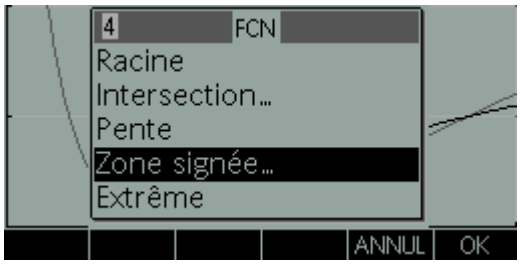




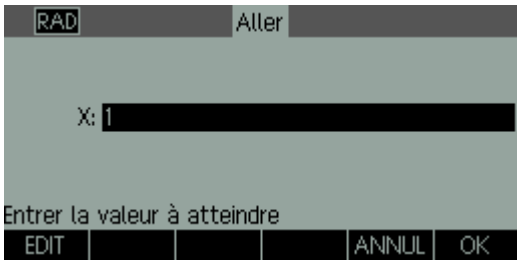


La HP 39gII permet de colorier et de calculer l'aire de la surface entre les deux courbes sur l'intervalle demandé. Pour cela, appuyer sur la touche [F4].

Choisir « Zone signée… ».

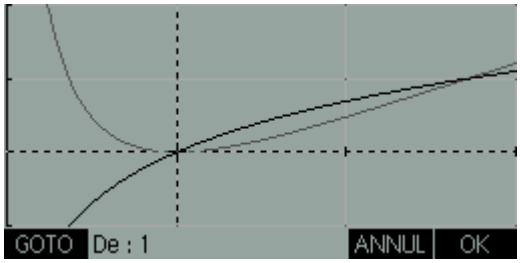


Placer le curseur en $x = 1$ en appuyant sur [F1] (GOTO) et en entrant 1 comme valeur pour x.

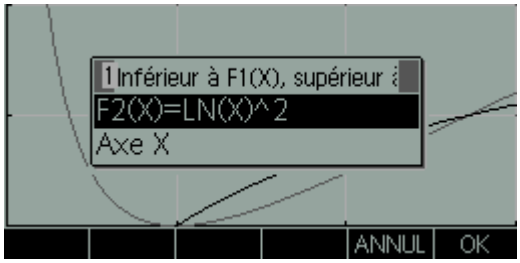


Calcul d'aire entre deux courbes

HP 39gII

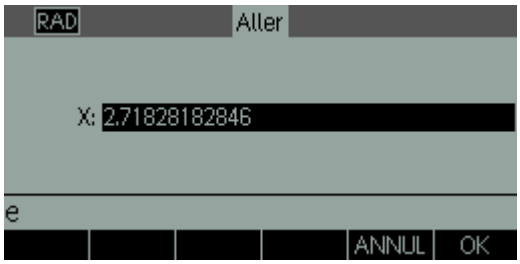


Appuyer sur [F6] pour valider et choisir F2(X).



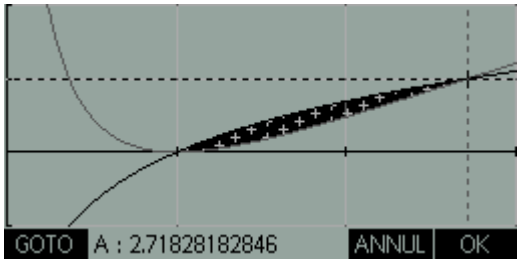
Placer le curseur en $x = e$ en appuyant sur [F1] (GOTO) et en entrant e comme valeur pour x. Pour entrer le symbole e, appuyer sur cette séquence de touches :

[SHIFT] [e^x LN H] [Clear]

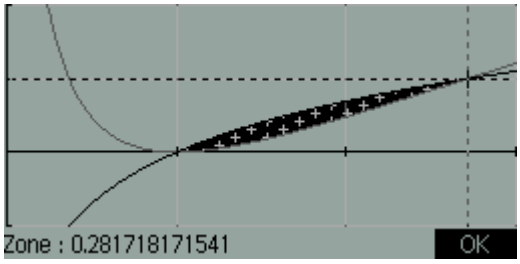


La surface entre les deux courbes se colore.

Appuyer sur [F6] pour valider.



La calculatrice renvoie alors une valeur pour l'aire de la zone.



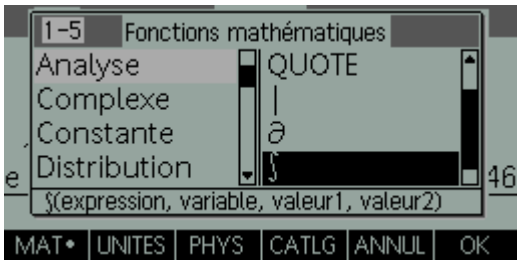
On peut vérifier cette valeur en calculant l'intégrale de la différence $f - g$ entre les bornes 1 et e qui correspond géométriquement à cette aire de la surface entre les deux courbes sur l'intervalle [1 ; e]. La position relative des deux courbes s'obtient avec le tableau de signes établi suivant :

x	0		1		e		$+\infty$
$\ln(x)$	‖	−	0			+	
$(1-\ln(x))$	‖	+		+	0	−	
$\ln(x)\times(1-\ln(x))$	‖	−	0	+	0	−	

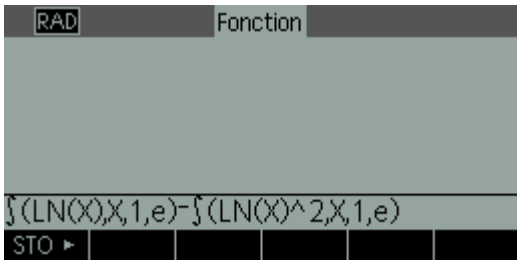
Calcul d'aire entre deux courbes

HP 39gII

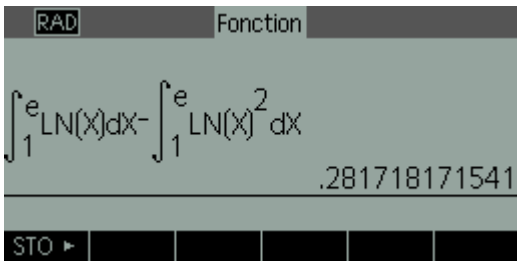
La courbe de f est au-dessus de celle de g sur l'intervalle d'étude.
Pour calculer une intégrale sur la HP 39gII, appuyer sur la touche [Home Modes] pour aller sur l'écran de calculs.
Aller chercher le signe intégrale dans le menu ANALYSE accessible depuis la touche [Math Cmds B].



Taper ensuite la différence des intégrales avec la syntaxe suivante ci-contre.



On retombe bien sur le résultat donné depuis l'écran graphique.



L'intégrale peut se calculer avec une intégration par parties sur l'intégrale de $\ln(x)$ puis en s'aidant de la fonction auxiliaire $G(x) = x\,(\ln(x)^2 - 2\ln(x) + 2)$.

Nombres complexes

HP 39gII

Soient Z1 = 3 +2i et Z2 = 1 – i deux nombres complexes.
1/ Calculer Z1 + Z2 ; Z1.Z2 et Z1/Z2.
2/ Donner le module et l'argument de Z1.

Touches d'accès au *i* complexe :

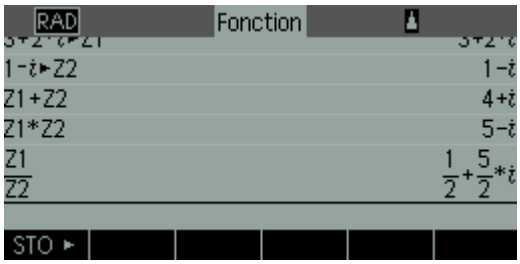
Solution pas à pas :

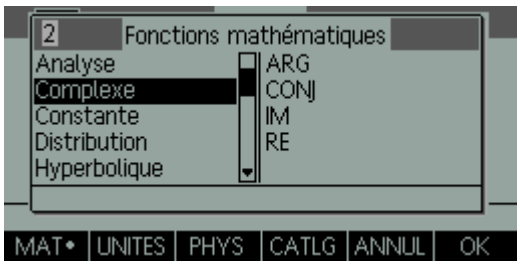
La HP 39gII peut stocker des nombres complexes dans les variables Z0 et Z9.

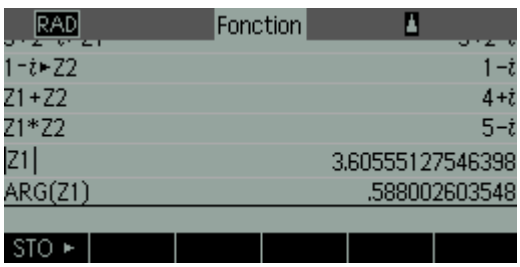
1/ On peut dès alors effectuer directement les calculs demandés sur Z1 et Z2.

2/ Depuis l'écran de calcul, appuyer sur la touche Math Cmds B pour accéder aux commandes sur les complexes.
L'argument s'obtient avec la commande ARG.
Le module s'obtient avec la commande ABS.

Captures d'écran :







Mesure principale d'un angle

1ère S

HP 39gII

1/ Déterminer la mesure principale d'un angle orienté donné à l'aide d'un algorithme.

2/ Tester l'algorithme avec 123π/4.

Solution pas à pas :

La mesure principale d'un angle se trouve dans l'intervalle]-π ; π].
On ajoutera ou en soustraira donc à la mesure d'angle donné les multiples successifs de 2π jusqu'à ce que l'intervalle soit atteint.

Pour ne pas être gêné à la sortie par du calcul non exact, le mieux est encore de considérer X comme une fraction P/Q facteur de π et de traiter P et Q.

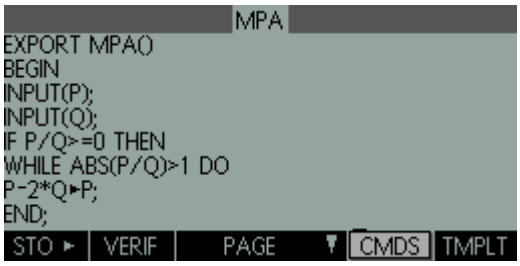
Entrée
Saisir P
Saisir Q
Traitement
Si P/Q≥0 Alors
Tant que ABS(P/Q)>1
P prend la valeur P+2Q
Fin tant que
Sinon
Tant que ABS(P/Q)>1
P prend la valeur P-2Q
Fin tant que
Fin si
Sortie
Afficher P/Q.π

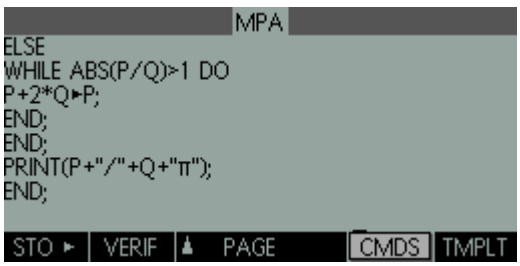
Le P+2Q vient de P/Q.π +2π = (P+2Q).π/Q
De même pour P-2Q.
Pour la sortie, pour échapper au calcul non exact, on affiche le / et le π en chaînes de caractères.

Le programme retourne parfaitement la mesure principale de 123π/4.

Captures d'écran :

On traduira alors cet algorithme sur HP 39gII :





Approximation de racines carrées

HP 39gII

Ts

Objectifs : approcher la valeur d'une racine carrée avec une suite récurrente, écrire un algorithme.
Mots-clés : suite, récurrence, algorithme, racine carrée.

Enoncé : On considère l'algorithme suivant pour approximer la racine carrée d'un nombre X :

- On choisit un nombre de départ Y.
- On calcule la demi-somme de Y et de X/Y.
- On affecte ce résultat à Y et on recommence.

Faire tourner l'algorithme.
Associer l'algorithme à une suite qui tend vers $\sqrt{X}$.

Solution pas à pas :

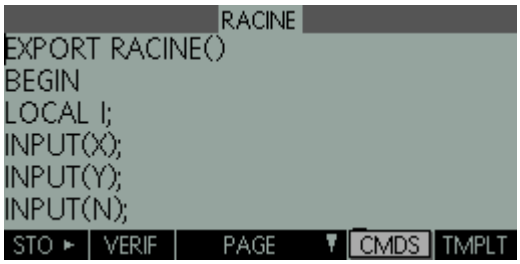
On peut commencer par écrire l'algorithme sous forme générique :

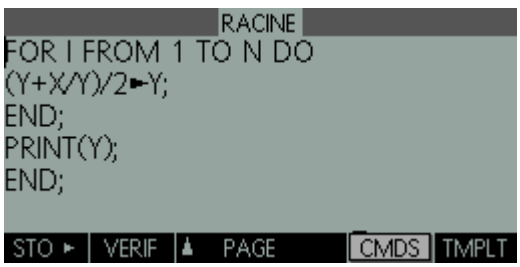
Variables :
X (dont on veut approcher la racine carrée)
Y nombre de départ
N (nombre d'itérations)
I (compteur)
Entrées :
Demander X
Demander Y
Demander N (le nombre d'itérations à calculer)
Traitement :
Pour I allant de 1 à N faire
Affecter (Y+X/Y)/2 à Y
Fin du pour
Sortie :
Afficher Y

Pour mettre cet algorithme sur la HP 39gII, appuyer sur les touches SHIFT et 1 Prgm X puis appuyer sur

F2 (NEW) pour créer un programme.
Nommer le programme RACINE et taper le code ci-contre.

Captures d'écran :





Approximation de racines carrées

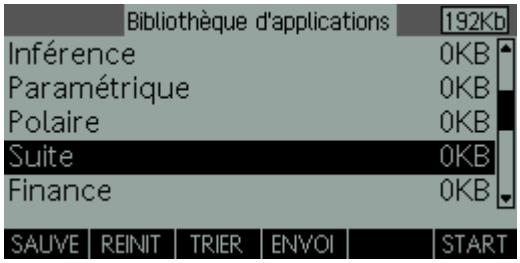
HP 39gII

En prenant X=2, Y=1 et N=100, on obtient :

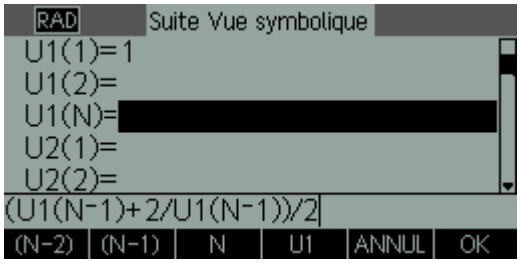
Soit une bonne approximation de $\sqrt{2}$.
L'algorithme ne fait en fait que calculer les termes de la suite suivante :

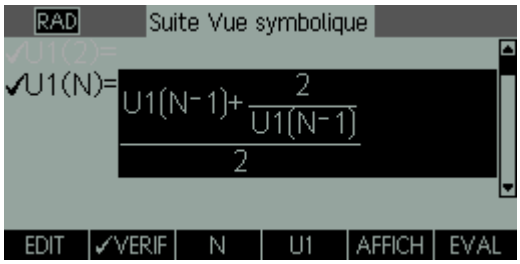
$$U_{n+1} = (U_n + X/U_n)/2 \text{ avec } U_0 = Y$$

On peut étudier la suite sur la HP-39gII en lançant l'application *Suite* depuis la touche Apps Info.



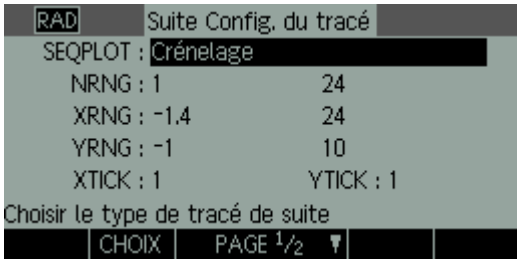
Rentrer le 1er terme Y (on prendra Y=1) dans U(1) puis l'expression de la suite récurrente dans U1(N) :





Régler le tracé en mode toile d'araignée. Pour cela, appuyer sur les touches SHIFT et Plot Setup.

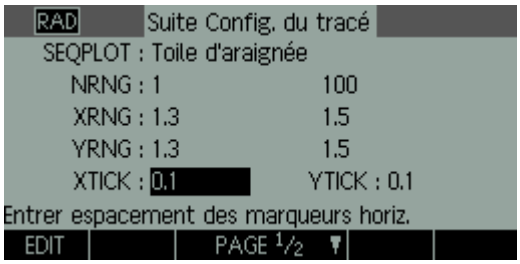
Appuyer ensuite sur F2 (CHOIX) pour choisir *Toile d'araignée*.



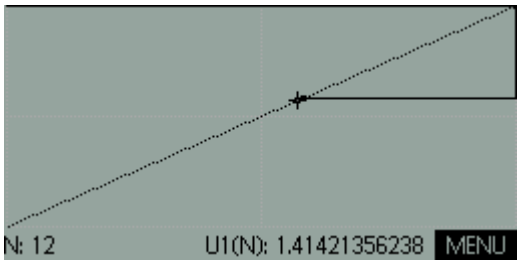
Approximation de racines carrées

HP 39gII

On règle les champs de la fenêtre (touches SHIFT Plot Setup).



On accède au tracé en appuyant sur Plot Setup .



La suite converge rapidement vers $\sqrt{2}$.
On peut le démontrer en posant $U_{n+1}=f(U_n)$.
Dans ce cas, $f(x)=1/2(x+2/x)$. Il suffit de résoudre l'équation $f(l)=l$.
Il vient $2l=l+2/l$ ou encore $l=2/l$ donc $l^2=2$. Donc $l=\sqrt{2}$ car U_0 et donc tous les termes sont positifs.
Remarque : on prendra le 1^{er} terme U_0 = Y non nul car sinon on obtient la suite constante nulle.

Théorème des restes chinois

HP 39gII

TS Spé

On se propose, dans cette question, de déterminer tous les entiers relatifs N tels que $\begin{cases} N \equiv 5[13] \\ N \equiv 1[17] \end{cases}$.

Solution pas à pas :

La HP 39 gII possède une commande qui résout directement ce genre de problème.

Elle est accessible depuis la touche [Math Cmds B] et se trouve dans le menu « Nombre entier ».

On tape alors ceci :

[SHIFT] [5 [U] [5 [U] [, Mem O] [1 Prgm X] [3 π Z]

[SHIFT] [6] V] [, Mem O] [SHIFT] [5 [U] [1 Prgm X]

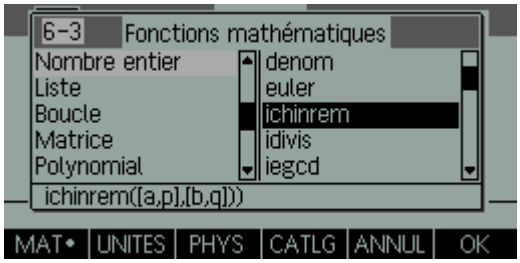
[, Mem O] [1 Prgm X] [7 List P] [SHIFT] [6] V] [) Paste M]

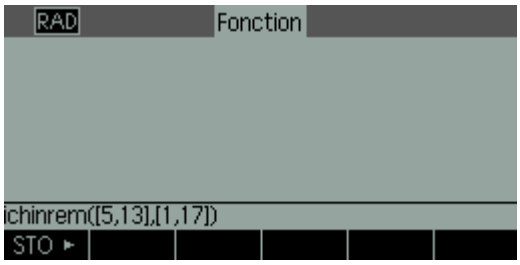
[ENTER ANS]

On obtient alors les solutions : tous les nombres entiers congrus à -203 modulo 221, c'est-à-dire congrus à 18 modulo 221.

Les questions suivantes permettent de le démontrer :

Captures d'écran :





a. Vérifier que 239 est solution de ce système.

b. Soit N un entier relatif solution de ce système. Démontrer que N peut s'écrire sous la forme $N = 1+17x = 5+13y$ où x et y sont deux entiers relatifs vérifiant la relation $17x - 13y = 4$.

c. Résoudre l'équation $17x - 13y = 4$ où x et y sont des entiers relatifs.

d. En déduire qu'il existe un entier relatif k tel que $N = 18 + 221k$.

e. Démontrer l'équivalence entre $N \equiv 18[221]$ et $\begin{cases} N \equiv 5[13] \\ N \equiv 1[17] \end{cases}$.

Intervalle de confiance

HP 39gII

TS

Un échantillon de 10 000 personnes sur une population donnée présente 7,5% de personnes à soigner pour un cholestérol important.
Donner un intervalle dans lequel on soit « sûr » à 95% de trouver le nombre exact de personnes à soigner sur les 10 000.

Solution pas à pas :

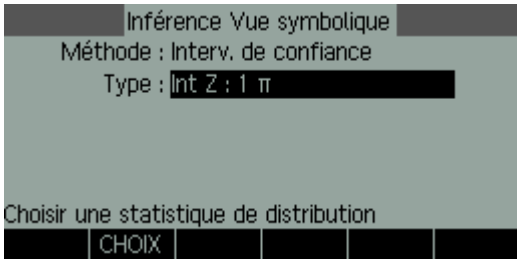
La HP 39gII possède les outils nécessaires pour directement obtenir l'intervalle de confiance recherché.
Lancer l'application « Inférence » depuis la touche [Apps Info].

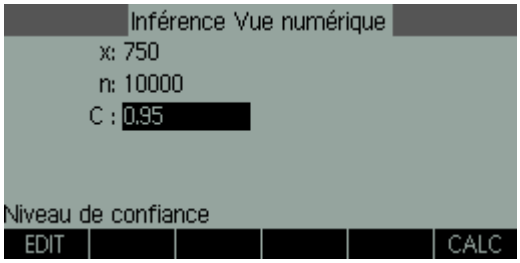
Appuyer sur la touche [Symb Setup] pour régler la méthode sur « Intervalle de confiance » et le type sur Int Z : 1 π

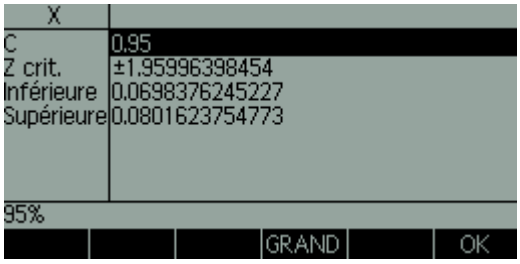
Appuyer sur la touche [Num Setup] pour rentrer les données de l'énoncé.
n est le nombre de personnes.
x est le nombre de personnes touchées par un cholestérol important : 0.075 x 10 000 = 750.
C est le niveau de confiance : 0.95.

Appuyer sur [F6] (CALC) pour obtenir l'intervalle recherché :
entre ≈0,0698×10 000 = 698 personnes
et ≈ 0,0802×10 000 = 802 personnes.

Captures d'écran :







Probabilité : loi normale

HP 39gII

TS

Exercice type : La température T au mois de juillet suit une loi normale de moyenne 22°C et d'écart type 4°C.

1/ Calculer la probabilité que la température soit inférieure à 19°C.
2/ Calculer la probabilité que la température soit supérieure à 27°C.
3/ Calculer la probabilité que la température soit comprise entre 24°C et 30°C.
4/ Trouver la température t telle que p(T≤t) = 0,8.
5/ Représenter graphiquement la densité *f* de T.
6/ Que représente p(30≤T≤35) sur le graphique ?

Solution pas à pas :

1/ La HP 39gII permet de calculer des probabilités avec la loi normale. Pour cela, il faut utiliser la commande **normald_cdf(** suivie des deux paramètres (moyenne m = 22 et écart type = σ = 4) pour une loi normale de paramètres $\mathcal{N}(m, \sigma^2) = \mathcal{N}(22,4^2)$ et de borne supérieure 19°C.
Ainsi pour calculer p(T≤19), on tape :

normald_cdf(22,4,19)

La probabilité qu'il fasse moins de 19°C au mois de juillet est de ≈0,23.

2/ p(T≥27) = 1 – p(T≤27)
On tape donc :

1 – normald_cdf(22,4,27)

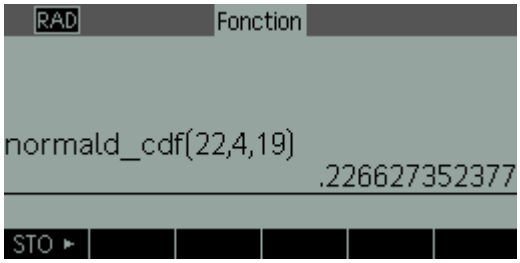
La probabilité qu'il fasse plus de 27°C au mois de juillet est de ≈0,11.

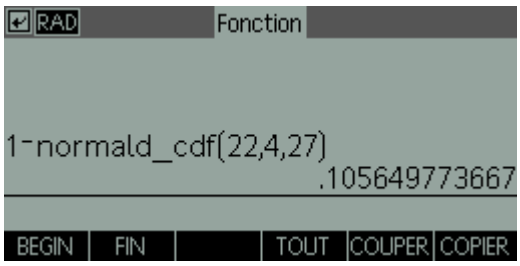
3/ p(24≤T≤30) = p(T≤30) – p(T≤24).
On tape donc :

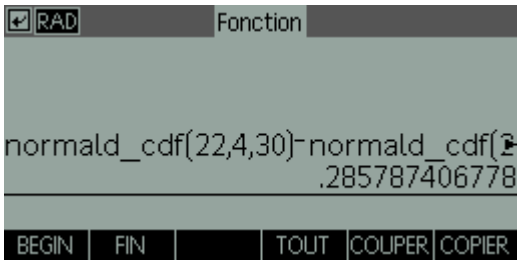
normald_cdf(22,4,30) – norrmal_cdf(22,4,24)

La probabilité qu'il fasse entre 24°C et 30°C au mois de juillet est de ≈0,29.

Captures d'écran :







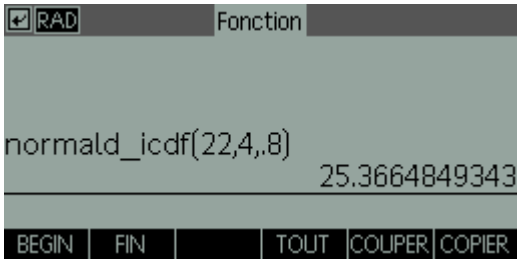
Probabilité : loi normale

HP 39gII

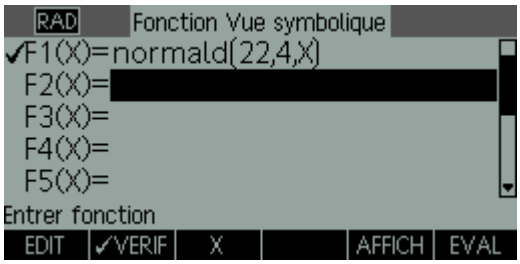
4/ On utilise la commande inverse **normald_icdf(**
On tape alors :

normald_icdf(22,4,0.8)

La probabilité p(T≤t) = 0,8 pour t ≈ 25,4°C.

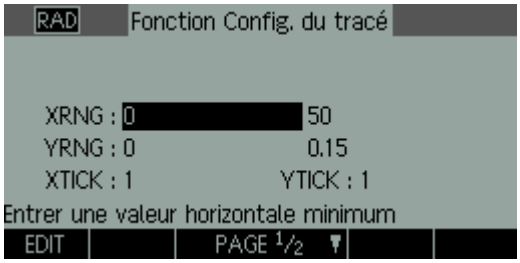


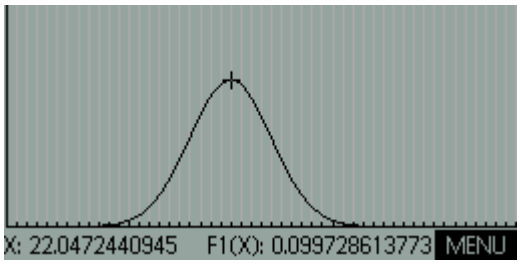
5/ La densité f de T est donnée par la commande **normald(**
f(x)=normald(22,4,x)
Dans l'application Fonction, on peut alors entrer F1(X)=normald(22,4,X)



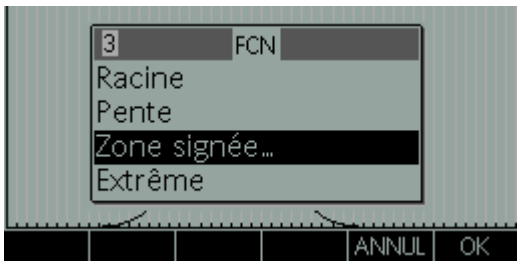
On règle la fenêtre d'affichage depuis SHIFT Plot Setup .

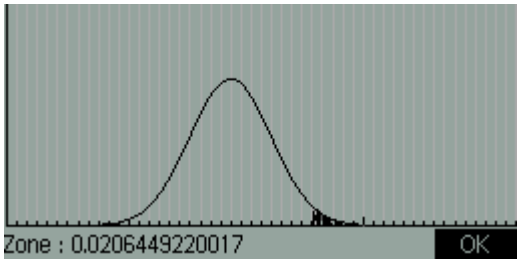
On pourra effectuer les réglages suivants pour apercevoir le graphique obtenu depuis la touche Plot Setup .





6/ La probabilité p(30≤T≤35) que la température soit comprise entre 30°C et 35°C est représentée graphiquement par l'aire sous la courbe entre les abscisses 30 et 35 (aire de la partie du plan délimitée par les droites d'équation $x = 30$, $x = 35$ et $\mathcal{C}_f$).
On peut le visualiser sur la HP 39gII en appuyant à l'écran graphique sur F6 (MENU), F4 (FNCT), en choisissant Zone signée...., en appuyant sur F1 (GOTO) pour entrer x =30, sur F6 (OK) et à nouveau sur F1 (GOTO) pour entrer x =35 pour finir sur F6 (OK).

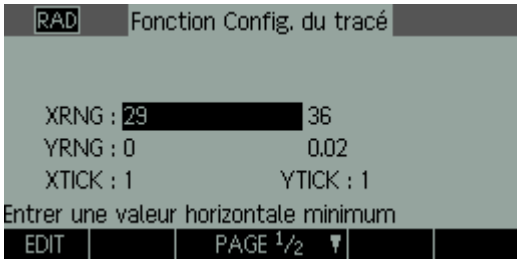


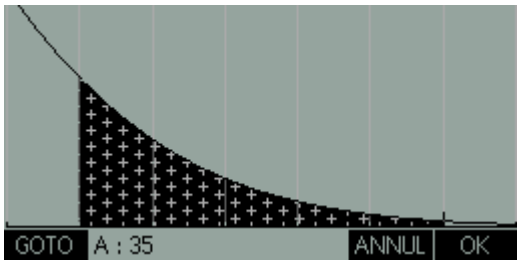


Probabilité : loi normale

HP 39gII

On peut régler la fenêtre pour mieux voir la zone hachurée.





Marche aléatoire

HP 39gII

Niveaux : Premières / Terminales S.
Objectifs : vérifier une conjecture, écrire et utiliser un algorithme.
Mots-clés : algorithme, itération, boucle While.

Enoncé : Un pion est placé sur la case départ du plateau ci-dessous :

| | | | | | pion | | | | |

Le lancer d'une pièce équilibrée détermine le déplacement du pion : PILE, le pion se déplace vers la droite ; FACE, le pion se déplace vers la gauche. À chaque lancer, on attribue le réel +1 si le résultat est PILE et -1 si le résultat est FACE.
Un trajet est une succession de *n* déplacements. La variable aléatoire Sn est la somme des nombres
1 ou -1 correspondant aux *n* lancers d'un trajet.
On s'intéresse à l'évènement D_n : « le pion est revenu à la case départ après les *n* déplacements d'un trajet ».

L'algorithme ci-dessous permet de réaliser la simulation d'un trajet de *n* déplacements, la valeur de *n* pouvant être choisie par l'utilisateur.

Variables :

N,S,A,I : nombres réels

Traitement :

```
Saisir N
S prend la valeur 0
Pour I variant de 1 à N
   A prend la valeur d'un entier aléatoire 0 ou 1
   Si A=1
      Alors S prend la valeur S+1
Sinon S prend la valeur S-1
   Fin Si
Fin Pour
```

Sortie :

```
Afficher S
Fin
```

1/ Utiliser cet algorithme, sur la calculatrice, pour réaliser plusieurs simulations dans le cas où le pion effectue 1 ou 2 déplacements.

2/ Modifier l'algorithme précédent de façon à pouvoir simuler plusieurs trajets du pion et calculer la fréquence de l'évènement D_n.

Marche aléatoire

HP 39gII

Solution pas à pas :

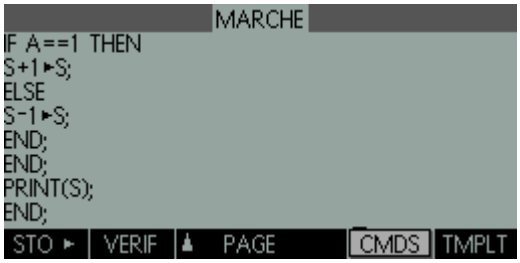
1/ On adapte l'algorithme sur la HP 39gII.

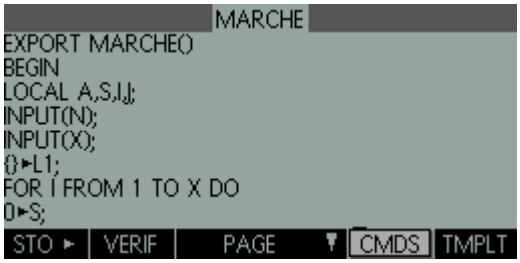
Saisir 1 ou 2 pour N quand on fait tourner l'algorithme.
Le programme retourne la variable aléatoire Sn qui correspond aussi à la position du pion (0 pour la case départ, +1 pour 1 case après la case de départ, -2 pour 2 cases avant la case départ, etc...).

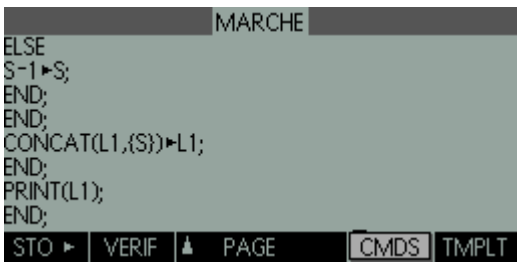
2/ Il faut lancer plusieurs fois l'algorithme précédent pour simuler plusieurs trajets.
On stockera chaque trajet dans une liste ou affichera successivement chaque valeur de S.
On lancera alors l'algorithme suivant :

Variables :
X,I : nombres entiers
Traitement :
Saisir X (nombre de trajet à simuler)
L est déclarée comme liste vide
Pour I variant de 1 à X
Lancer le programme MARCHE
Ajouter S comme élément à la liste L
Fin Pour
Sortie :
Afficher L
Fin

Captures d'écran :



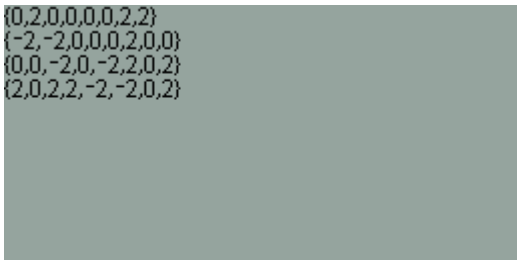




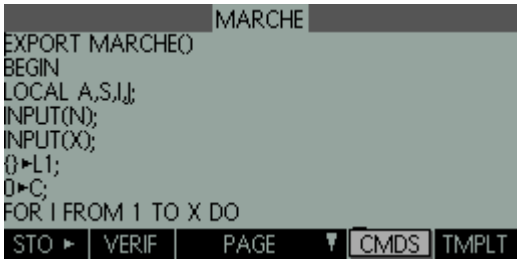
Marche aléatoire

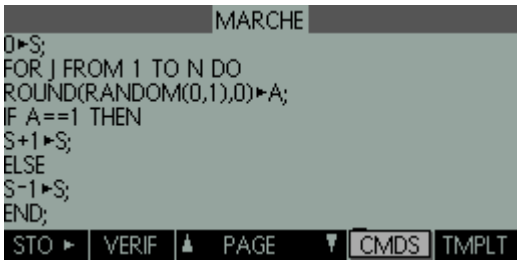
HP 39gII

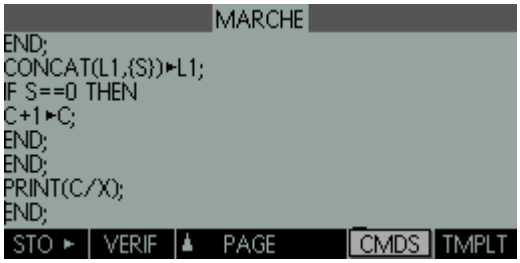
Le programme retourne la liste des cases d'arrivée du pion (ici, des résultats de 8 simulations de 2 déplacements).



Pour obtenir la fréquence de l'évènement D_n, on divise le nombre de 0 de la liste par le nombre d'éléments de la liste.
On incrémente un compteur pour dénombrer les 0.







Matrices

HP 39gII

TS Spé

Exercice $A=\begin{pmatrix}1 & 3\\ 4 & 2\end{pmatrix}$ et $Q=\begin{pmatrix}1 & 1\\ -4 & 3\end{pmatrix}$

1) Déterminer la matrice P, inverse de Q, à l'aide de la calculatrice.
2) Calculer $Q\times A\times P$.
3) **En déduire** la matrice A^n , pour tout entier naturel non nul n.

Solution pas à pas :

La HP 39 gII permet d'effectuer du calcul matriciel.
On définit une matrice entre crochets. A l'intérieur de ces crochets, on écrit chaque ligne de la matrice entre crochets.
On stocke chaque matrice dans les variables M1 et M2.

1) On peut alors directement calculer Q^{-1} avec les touches $\sqrt[y]{}$ x^y K , (−) ABS ; , 1 Prgm X .

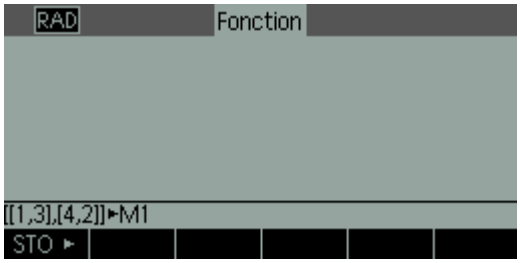
Appuyer sur la touche $a\leftrightarrow b/c$ pour obtenir des coefficients exacts dans la matrice inversée.

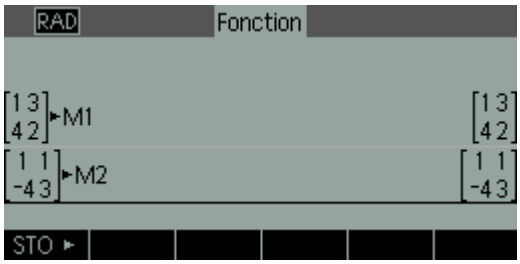
$Q^{-1}=\begin{pmatrix}\frac{3}{7} & \frac{-1}{7}\\ \frac{4}{7} & \frac{1}{7}\end{pmatrix}$

2) Le résultat du produit demandé est :
$\begin{pmatrix}5 & 0\\ 0 & -2\end{pmatrix}$
Il s'agit d'une matrice diagonale.

3) $\begin{pmatrix}5 & 0\\ 0 & -2\end{pmatrix}^n=\begin{pmatrix}5^n & 0\\ 0 & (-2)^n\end{pmatrix}$

Captures d'écran :





Matrices
HP 39gII

Comme P.Q = Identité,

Q.A.P.Q.A.P = Q.A.A.P = Q.A².P = $\begin{pmatrix} 5^2 & 0 \\ 0 & (-2)^2 \end{pmatrix}$

Q.A².P.Q.A.P = Q.A².A.P = Q.A3.P = $\begin{pmatrix} 5^3 & 0 \\ 0 & (-2)^3 \end{pmatrix}$

etc…

On arrive à Q.Aⁿ.P = $\begin{pmatrix} 5^n & 0 \\ 0 & (-2)^n \end{pmatrix}$.

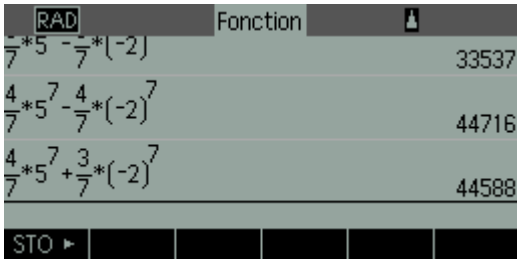
Donc Aⁿ = P. $\begin{pmatrix} 5^n & 0 \\ 0 & (-2)^n \end{pmatrix}$.Q

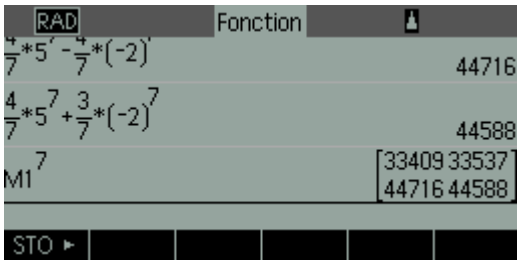
$$\mathbf{A^n} = \begin{pmatrix} \frac{3}{7}.\mathbf{5^n} + \frac{4}{7}.(-\mathbf{2})^{\boldsymbol{n}} & \frac{3}{7}.\mathbf{5^n} - \frac{3}{7}.(-\mathbf{2})^{\boldsymbol{n}} \\ \frac{4}{7}.\mathbf{5^n} - \frac{4}{7}.(-\mathbf{2})^{\boldsymbol{n}} & \frac{4}{7}.\mathbf{5^n} + \frac{3}{7}.(-\mathbf{2})^{\boldsymbol{n}} \end{pmatrix}$$

Avec cette formule,

$A^7 = \begin{pmatrix} 33\,409 & 33\,537 \\ 44\,716 & 44\,588 \end{pmatrix}$

On vérifie…





Capacité calorifique : méthode des mélanges

HP 39gII

Durée : 1 heure
Matériel : HP 39gII, HP StreamSmart, sonde de température, récipient, eau.

Objectif : Déterminer la capacité calorifique d'un récipient.
Expérience : Dans le récipient, verser un volume V1 d'eau à température ambiante.
Noter la masse m1 et la température T1 de cette eau.
Dans le récipient, verser rapidement ensuite un volume V2 d'eau chaude.
Noter la masse m2 et la température T2 de cette eau.
Couvrir et observer l'évolution de la température jusqu'à l'équilibre T3.

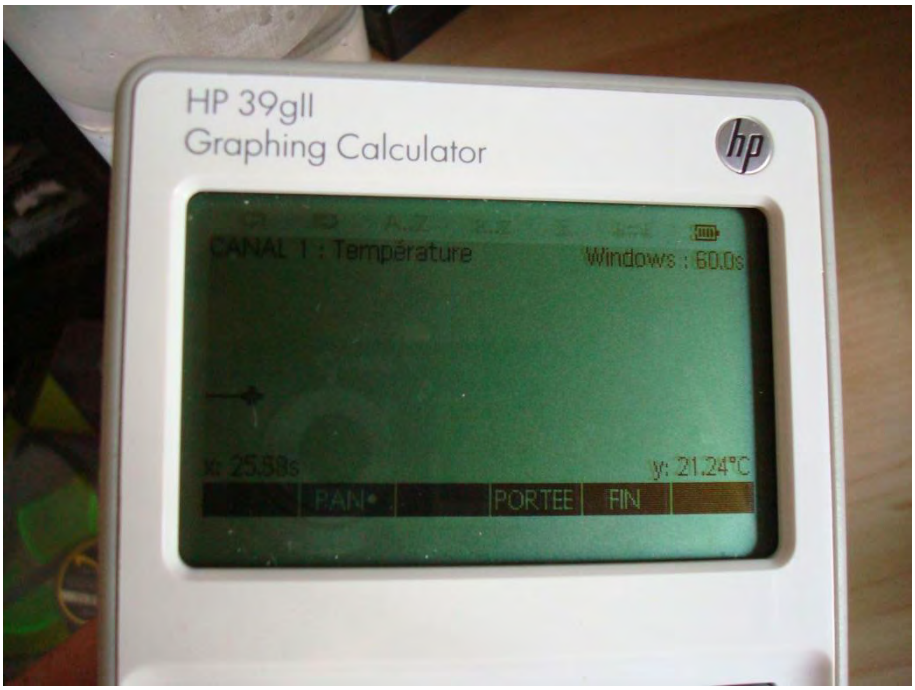
Solution pas à pas :

Dans notre expérience, on verse 10 cL (m_2 = 0,1 kg) d'eau chaude à T_2 = 43,22°C dans 10 cL (m_1 = 0,1 kg) d'eau froide à T_1 = 21,24°C contenue dans un cul de poule Ikea.

Capacité calorifique : méthode des mélanges

HP 39gII

Pour mesurer la température en fonction du temps sur la HP 39gII, il faut relier la calculatrice au boitier StreamSmart sur lequel est branchée la sonde de température.
L'application DataStreamer sert d'interface de récupération des données.



Une pression sur la touche F5 permet de lancer la mesure.

Une courbe se trace alors en temps réel (ici température en fonction du temps).

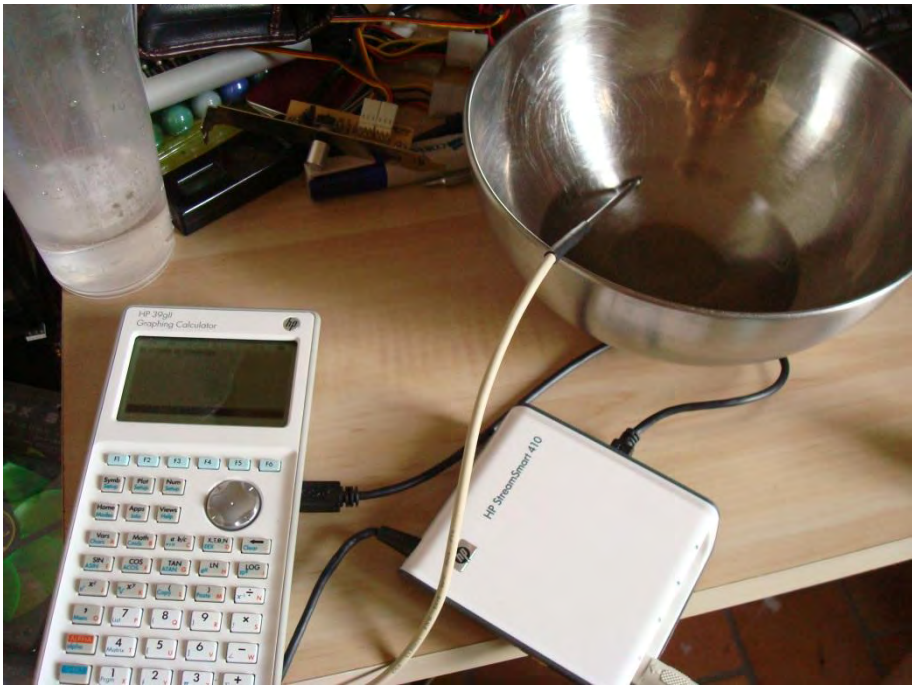
Une nouvelle pression sur la touche F5 arrête la mesure.

Il est possible d'exporter le graphique pour l'exploiter ensuite mathématiquement.

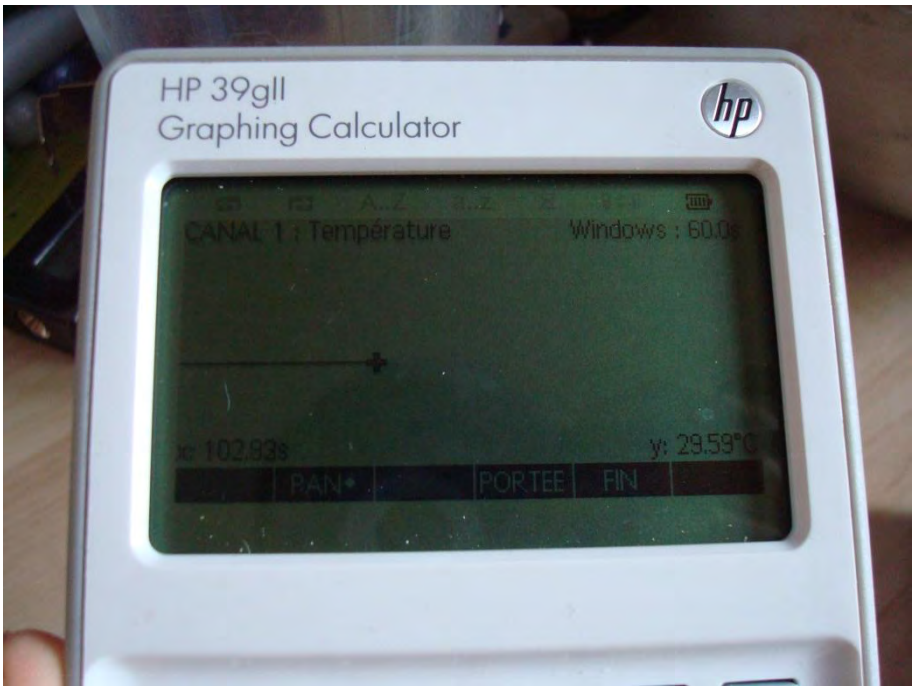
Capacité calorifique : méthode des mélanges

HP 39gII

On plonge la sonde de température dans le cul de poule et on observe l'évolution de la température jusqu'à une stabilisation à T_3.



Après mélange des deux eaux et recouvrement du récipient, on observe une stabilisation de la température vers T_3 = 29,59°C.



La loi de conservation de l'énergie indique que la somme des énergies transférées par l'eau froide, le récipient et l'eau chaude est nulle.

Capacité calorifique : méthode des mélanges

HP 39gII

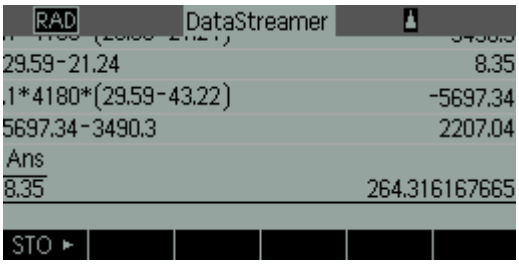
Donc :

$$Q_1 + Q_{cal} + Q_2 = 0 \quad \text{soit} \quad m_1 c_{eau}.(T_3 - T_1) + C.(T_3 - T_1) + m_2 c_{eau}.(T_3 - T_2) = 0$$

Avec $c_{eau} = 4180\ J.K^{-1}.kg^{-1}$

On complète l'équation avec les valeurs expérimentales trouvées :

0,1×4180×(29,59 – 21,24) + C.(29,59 – 21,24) + 0,1×4180×(29,59 – 43,22) = 0
c'est-à-dire : 3490,3 + 8,35C – 5697,34 = 0
8,35C = 2207,04
C≈ 264,32 J/K



La capacité calorifique de notre récipient est d'environ **264,32 J/K**.