HIGHWAY CONSTRUCTION

PROGRAMS

FOR USE WITH THE HEWLETT-PACKARD 28S CALCULATOR

WRITTEN BY S.R. COVINGTON

HEWLETT-PACKARD 28 PROGRAMS
TABLE OF CONTENTS

INTRODUCTION

These HP28 programs were written primarily for Highway Construction field data computations. Following you will find some tips pertaining to the operation of these specific programs. Please refer to your HP28S owner's manual for instructions on operating your HP28S calculator.

Some programs have infinite loops, to halt these programs you must type "KILL" then press the enter key. All other programs halt after the output has been displayed. With all of these programs you may want to add the "purg" sub-program to clear the registers when you complete your calculations, this will extend the battery life in your calculator.

All programs have example problems. It is always a good idea to run through these examples after you have input a program into your calculator to check for bugs. These examples have been checked for there accuracy so if you get erroneous answers you have input the program incorrectly, go back and check it for mistakes. These examples were written with the idea that those who use them have experience manipulating the data to achieve the correct results.

For those using the HP28S, it is advised that you organize the HP28S's R.A.M. in the same manner that a hard drive for a micro computer is organized. Using this technique, you will not clutter up the "Home Directory" with all of your programs and data. Start with "civil" and create a PARENT directory. Then change from the Home Directory to the "civil" parent directory that you just created. From the "civil" parent directory create sub-directories for each kind of program. Some Examples could be, "AREA", "GRADE", and "BERG". Then go into the sub-directory to input the program.

I have found the "ORDER" command on page 143 of the reference manual useful for reorganizing my directories. After the programs have been inputed you may wish to rearrange them, this command will accomplish this for you. There is also a short program on page 55 of the owner's manual called "RENAME" that I have found useful. I put this program on the "HOME" directory so that it can be accessed from the other directories, it is useful to rename variables and programs.

These programs do not include printer control for the HP 82240A INFRARED PRINTER. Printer control may be included in one of two ways. The first is to replace the "WAIT" statement in the program structure with "PRLCD", then put the HP28S in "TRACE" mode, in this manner you will have a print out of both the prompts and the results. The second way is to recall stored results on to the stack and print them with the "PR1" command in the program structure.

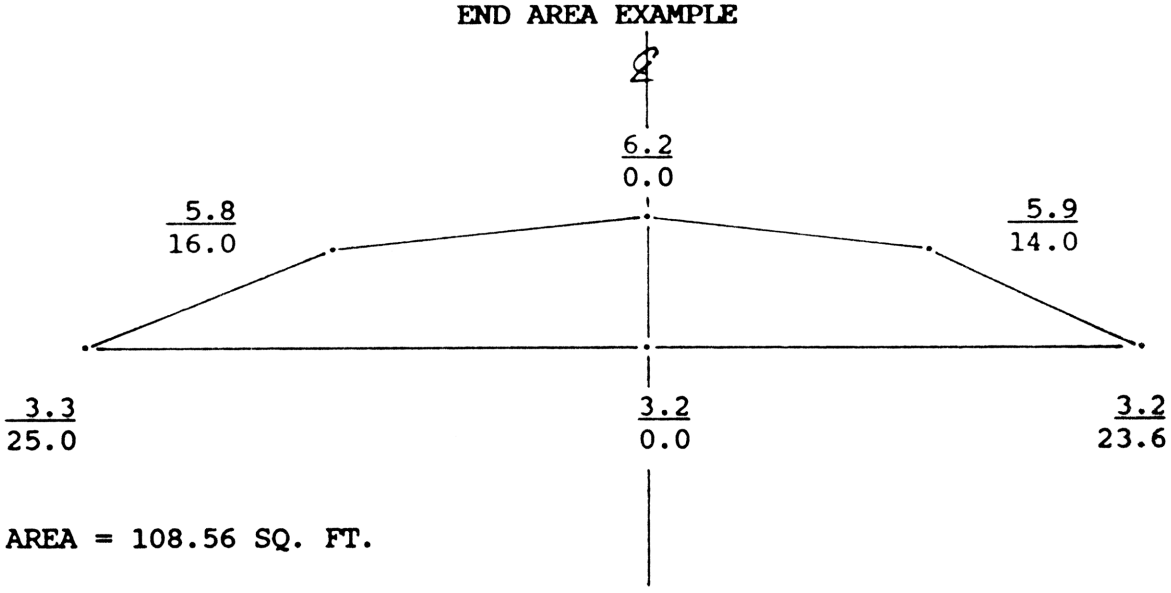
Before using any of these programs you should read all the notes at the bottom of the page. Some programs have unique steps for the input of data, to avoid confusion read all notes first.

END AREA EARTH WORK

```
<< "ROD" HALT DUP 'a' STO 'b' STO DROP "DISTANCE" HALT DUP 'c' STO
'd' STO DROP 0 'g' STO EA1 >>

# ENTER 'AREA' STO
```

```
<< "ROD" HALT 'j' STO DROP "DISTANCE" HALT 'f' STO DROP '(j-a)*
(c+f)+g' EVAL 'g' STO 'j' RCL 'a' STO 'f' RCL 'c' STO IF a b == c d
== AND THEN g 2 / 'h' STO CLLCD "DBL END AREA" 1 DISP 'h' RCL 2 FIX
3 DISP 4 WAIT { a b c d j f g h } PURGE STD CLMF ELSE CLMF EA1 END
>>

# ENTER 'EA1' STO
```

END AREA EXAMPLE



AREA = 108.56 SQ. FT.

* NOTE: ALWAYS ENTER DISTANCES LEFT OF CENTER LINE AS NEGATIVE.

END AREA AND VOLUME EARTH WORK

```
<< 0 DUP 'h' STO 'k' STO EV1 >>

# ENTER 'ENDVOL' STO


<< "ROD" HALT DUP 'a' STO 'b' STO DROP "DISTANCE" HALT DUP 'c' STO
'd' STO DROP 0 'g' STO EV2 >>

# ENTER 'EV1' STO


<< "ROD" HALT 'j' STO DROP "DISTANCE" HALT 'f' STO DROP '(j-a)*
(c+f)+g' EVAL 'g' STO 'j' RCL 'a' STO 'f' RCL 'c' STO IF a b == c d
== AND THEN g 2 / 'g' STO CLLCD "DOUBLE END AREA" 1 DISP 'g' RCL 2
FIX 3 DISP 4 WAIT CLMF 2 FIX "INTERVAL" HALT 'l' STO DROP CLLCD
"VOLUME" 1 DISP '(g+h)*l/54' EVAL DUP k + 'k' STO 3 DISP 4 WAIT
"ACCUM. VOLUME" 1 DISP 'k' RCL 3 DISP 4 WAIT 'g' RCL 'h' STO { a b c
d f j } PURGE STD CLMF EV1 ELSE CLMF EV2 >>

# ENTER 'EV2' STO


<< { g h k l } PURGE >>

# ENTER 'EVPR' STO
```

ENDVOL EXAMPLE

```
2+15  00 SECTION                                    A=0       V=0


2+45  1.0  1.9 12.3 18.6 20.0 23.6 26.0 27.2  A=356.5   V=198.03
     14.0 16.0 33.0 42.0 53.0 61.0 75.0 92.6


2+72 00 SECTION                                      A=0       V=178.23
                                              ACCUM. VOL.=376.25
```

* NOTE: 1) ROD & DISTANCE FOR BOTH 00 SECTIONS HAVE TO BE ENTERED AS 0 TWICE. THE INTERVAL IS THE DISTANCE BETWEEN STATIONS.

2) ALWAYS ENTER DISTANCES LEFT OF CENTER LINE AS NEGATIVE.

VOLUME EARTH WORK

```
<< "STATION" HALT 'd' STO DROP "AREA" HALT 'f' STO DROP 0 'a' STO
VOL1 >>

# ENTER 'VOLUME' STO


<< "STATION" HALT 'b' STO DROP "AREA" HALT 'c' STO DROP CLLCD 2 FIX
"VOLUME" 1 DISP '(b-d)*(f+c)/54' EVAL DUP a + 'a' STO 3 DISP 4 WAIT
"ACCUM. VOLUME" 1 DISP 'a' RCL 3 DISP 4 WAIT 'b' RCL 'd' STO 'c' RCL
'f' STO STD { b c } PURGE CLMF VOL1 >>

# ENTER 'VOL1' STO


<< { a b f } PURGE >>

# ENTER 'VPUR' STO
```

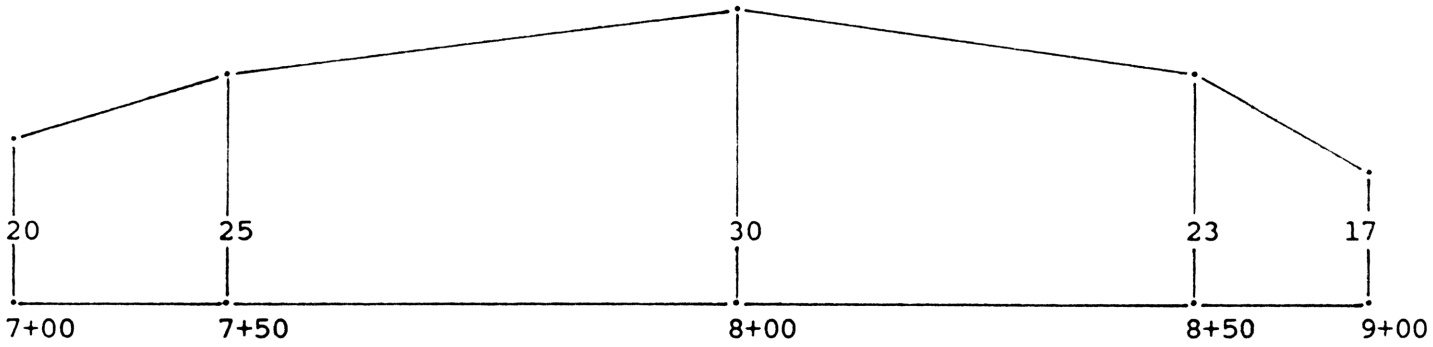
VOLUME EXAMPLE

	STATION	DISTANCE	AREA	DBL. AREA	ACCUM. VOL.
00 SEC.	2+15		0		
		30		356.5	198.1
	2+45		356.5		
		27		356.5	178.3
00 SEC.	2+72		0		
				TOTAL	376.4 CU. YD

SEEDING AREAS

```
<< 0 DUP 'g' STO 'h' STO SEED1 >>

# ENTER 'SEED' STO


<< "STATION" HALT 'a' STO DROP "WIDTH" HALT 'b' STO DROP SEED2 >>

# ENTER 'SEED1' STO


<< "STATION" HALT 'c' STO DROP "WIDTH" HALT 'd' STO DROP '(c-a)*
((b+d)/2)' EVAL DUP 'f' STO h + 'h' STO CLLCD 4 FIX "AREA SQ. FT." 1 DISP
'f' RCL 3 DISP 4 WAIT "ACCUM AREA" 1 DISP 'h' RCL 3 DISP 4 WAIT 'f' RCL
43560 / 'f' STO "ACRES" 1 DISP 'f' RCL 3 DISP 4 WAIT 'f' RCL g + 'g' STO
"ACCUM ACRES" 1 DISP 'g' RCL 3 DISP 4 WAIT 'c' RCL 'a' STO 'd' RCL 'b'
STO STD CLMF SEED2 >>

# ENTER 'SEED2' STO


<< { a b c d f g h } PURGE >>

# ENTER 'PURG' STO
```

EXAMPLE

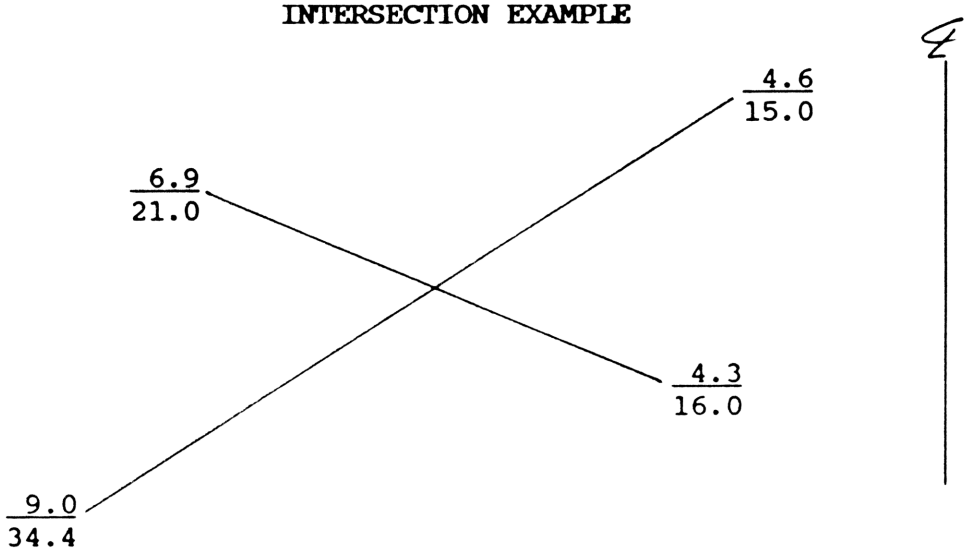


AREA SQ. FT.	1125.00	1375.00	1325.00	1000.00
ACCUM. AREA	1125.00	2500.00	3825.00	4825.00
ACRES	0.0258	0.0316	0.0304	0.0230
ACCUM. ACRES	0.0258	0.0574	0.0878	0.1108

INTERSECTION OF TWO STRAIGHT LINES

```
<< { STO r1 d1 r2 d2 r3 d3 r4 d4 } MENU HALT 001 >>

# ENTER '00' STO


<< '(r2-r1)/(d2-d1)' EVAL 'a' STO '(r4-r3)/(d4-d3)' EVAL 'b' STO
'(d3-d1)*a+r1' EVAL 'c' STO '(r3-c)/(a-b)' EVAL 'd' STO IF d 0 <
THEN 'd' SNEG END 'd+d3' EVAL 'f' STO '(f-d3)*b+r3' EVAL 'h' STO
CLLCD 2 FIX "00 ROD" 1 DISP 'h' RCL 3 DISP 4 WAIT "00 DISTANCE" 1
DISP 'f' RCL 3 DISP 4 WAIT { r1 d1 r2 d2 r3 d3 r4 d4 } PURGE STD
CLMF >>

# ENTER '001' STO
```

INTERSECTION EXAMPLE



NO.	1	2	3	4
ROD	4.6	9.0	4.3	6.9
DIST	15.0	34.4	16.0	21.0

```
00 ROD = 5.2
00 DISTANCE = 17.8
```

* NOTE: 1) ALWAYS INPUT THE SHOT CLOSEST TO THE CENTER LINE FIRST.

2) r1 = ROD 1 AND d1 = DISTANCE 1.

3) PRESS THE NEXT KEY TO ENTER r4 & d4 THEN PRESS SHIFT CONTINUE TO COMPLETE INPUT.

SIMPLE CURVE DATA AND DEFLECTIONS

```
<< "DEGREE OF CURVE" HALT HMS-> 'n' STO DROP "PI STATION" HALT 'o'
STO DROP "DELTA ANGLE" HALT HMS-> 'p' STO DROP CLLCD '5729.578/n'
EVAL 'q' STO "RADIUS" 1 DISP  'q' RCL 2 FIX 3 DISP 4 WAIT 'p/n*100'
EVAL 'r' STO "LENGTH OF CURVE" 1 DISP 'r' RCL 3 DISP 4 WAIT
'q*TAN(P/2)' EVAL 's' STO "TANGENT" 1 DISP 's' RCL 3 DISP 4 WAIT
"EXTERNAL" 1 DISP 'q*(1/COS(p/2)-1)' EVAL 3 DISP 4 WAIT 'o-s' EVAL
't' STO "PC STATION" 1 DISP 't' RCL 3 DISP 4 WAIT "PT STATION" 1
DISP 't+r' EVAL 3 DISP 4 WAIT STD { o p q r s } PURGE CLMF DEFL >>

# ENTER 'DATA' STO


<< "STATION" HALT 'u' STO DROP CLLCD "DEFLECTION" 1 DISP '(u-
t)*n/200' EVAL ->HMS 4 FIX 3 DISP 4 WAIT STD CLMF DEFL >>

# ENTER 'DEFL' STO


<< { n t u } PURGE >>

# ENTER 'PURG' STO
```

EXAMPLE

STATION	PT	DEFLECTION	CURVE DATA
29+63.11	PT	08° 00'00"	Dc 11° 30'00"
			PI 28+94.00
+50		07° 14'46"	DELT 16° 00'00"
+40	P.O.C.	06° 40'16"	
			R = 498.22'
+25		05° 48'31"	Lc = 139.13'
			T = 70.02'
29+00		04° 22'16"	Ex = 4.90'
+75		02° 56'01"	
+50		01° 29'46"	
28+23.98	PC	00° 00'00"	

SIMPLE CURVE CHORD LENGTH

```
<< "INST. STATION" HALT 'a' STO DROP "DEGREE OF CURVE" HALT HMS->
'b' STO DROP CLEN >>

# ENTER 'sccl' STO


<< "POINT ON CURVE" HALT "c" STO DROP '(c-a)*b' EVAL 'd' STO
'SIN(d/100)*(18E3/pi/b)/SIN(90-(d/200))' EVAL >>

# ENTER 'clen' STO
```

EXAMPLE

INSTRUMENT STATION	45+12.8
DEGREE OF CURVE	08° 00'00"
P.O.C.	49+22.2
CHORD LENGTH	403.85

* NOTE: 1) PRESS THE CLEN SOFT KEY TO CONTINUE TO ENTER SATIONS TO BE FIGURED FOR CHORD LENGTHS.

2) pi IS KEYED IN AS ↑π ON CALCULATOR.

SPIRAL CURVE DATA AND DEFLECTIONS

```
<< "PI STATION" HALT 'o' STO DROP "DEGREE OF CURVE" HALT HMS-> 'a'
STO DROP "LENGTH OF SPIRAL" HALT 'b' STO DROP "DELTA ANGLE" HALT
HMS-> 'c' STO DROP c 2 / 'z' STO a 200 / 'u' STO u b *  'd' STO
CLLCD "THETA s" 1 DISP 'd' RCL ->HMS 4 FIX 3 DISP 4 WAIT d *pi * 180
/ 't' STO D 3 / 'f' STO c 2 - d * 'l' STO "DELTA c" 1 DISP 'l' RCL
->HMS 3 DISP 4 WAIT 2 FIX 100 c * a / 'm' STO "LENGTH OF CURVE" 1
DISP 'm' RCL 3 DISP 4 WAIT 'm' RCL RND 'm' STO '18E3/(*pi)/a' EVAL
'g' STO "RADIUS" 1 DISP 'g' RCL 3 DISP 4 WAIT 'b*(t/3-
t^3/42+t^5/1320)' EVAL 'h' STO "Ys 1 DISP 'h' RCL 3 DISP 4 WAIT
'b*(1-t^2/10+t^4/216-t^6/9360)' EVAL 'v' STO "Xs" 1 DISP 'v' RCL 3
DISP 4 WAIT 'h-g*(-1*COS(d)+1)' EVAL 'j' STO 'v-g*SIN(d)' EVAL 'k'
STO '(g+j)*TAN(z)+k' EVAL 'n' STO "Ts" 1 DISP 'n' RCL 3 DISP 4 WAIT
'n' RCL RND 'n' STO "Es" 1 DISP '(g+j)/COS(z)-g' EVAL 3 DISP 4 WAIT
"p" 1 DISP 'h-g*(1-COS(d))' EVAL 3 DISP 4 WAIT "K" 1 DISP 'v-
g*SIN(d)' EVAL 3 DISP 4 WAIT "LONG TANGENT" 1 DISP 'v-h/TAN(d)' EVAL
3 DISP 4 WAIT "SHORT TANGENT" 1 DISP 'h/SIN(d)' EVAL 3 DISP 4 WAIT
"TS STATION" 1 DISP o n - DUP 'p' STO 3 DISP 4 WAIT "SC STATION" 1
DISP p b + DUP 'q' STO 3 DISP 4 WAIT "CS STATION" 1 DISP q m + DUP
'r' STO 3 DISP 4 WAIT "ST STATION" 1 DISP r b + 's' STO 3 DISP 4
WAIT STD CLMF DEFL >>

# ENTER 'DATA' STO


<< "STATION" HALT 'a' STO DROP IF a q ≤ THEN DEFL1 END IF a r ≤ THEN
DEFL2 ELSE CLLCD s a - 'v' STO b v - 'n' STO "DEFL. FROM C.S." 1
DISP 'u*w-(w/b)^2*f' EVAL -> HMS 4 FIX 3 DISP 4 WAIT STD "DEFL. FROM
S.T." 1 DISP '(v/b)^2*f' EVAL ->HMS 4 FIX 3 DISP 4 WAIT STD CLMF
DEFL END >>

# ENTER 'DEFL' STO


<< CLLCD a p - 'v' STO b v - 'w' STO '(v/b)^2*f' EVAL 'd' STO 4 FIX
"DEFL. FROM T.S." 1 DISP 'c' RCL ->HMS 4 FIX 3 DISP 4 WAIT STD "DEFL
FROM S.C." 1 DISP "d' RCL ->HMS 4 FIX 3 DISP 4 WAIT STD CLMF DEFL >>

# ENTER 'DEFL1' STO


<< CCLCD "SIMPLE CURVE" 1 DISP "DEFLECTION" 2 DISP a q - u * ->HMS 4
FIX 4 DISP 4 WAIT CLMF STD DEFL >>

# ENTER 'DEFL2' STO


<< { a b c d f g h j k l m n o p q r s t u v w z } PURGE >>
# ENTER 'PURG' STO


* NOTE: pi IS KEYED IN AS π ON CALCULATOR
```

SPIRAL CURVE EXAMPLE

STATION	CS	ST
ST 21+46.06	4° 40'00"	0° 00'00"
20+26.06	3° 45'24"	0° 12'36"
19+86.06	3° 21'36"	0° 22'24"
19+46.06	2° 55'00"	0° 35'00"
19+06.06	2° 25'36"	0° 50'24"
18+26.06	1° 18'24"	1° 29'36"
17+86.06	0° 40'36"	1° 53'24"
	0° 00'00"	2° 20'00"
CS 17+46.06	6° 49'58"	
17+00	6° 01'36"	
+50	5° 09'06"	
16+00	4° 16'36"	
+50	3° 24'06"	
15+00	2° 31'36"	
+50	1° 39'06"	
14+00	0° 46'36"	
	0° 00'00"	
SC 13+55.62	2° 20'00"	0° 00'00"
+95.62	0° 50'24"	2° 25'36"
11+15.62	0° 22'24"	3° 21'36"
+75.62	0° 12'36"	3° 45'24"
10+35.62	0° 05'36"	4° 06'24"
+95.62	0° 01'24"	4° 24'36"
TS 9+55.62	0° 00'00"	4° 40'00"

EXAMPLE

PI STATION	15+59.60	THETA Os	07° 00'00"
DEGREE OF CURVE	3° 30'00"	DELTA c	13° 39'55"
LENGTH OF SPIRAL	400.00'	Lc	390.44'
DELTA ANGLE	27°39'55"	Rc	1637.02'
		Ys	16.27'
		Xs	399.40'
		Ts	603.98'
		Es	53.09'
		P	4.0679'
		K	199.8974'
		LONG TAN	266.88'
		SHORT TAN	133.52'
		T.S. STATION	9+55.62
		S.C. STATION	13+55.62
		C.S. STATION	17+46.06
		S.T. STATION	21+46.06

SPIRAL CURVE OFFSETS

```
<< "LENGTH OF SPIRAL" HALT 'l' STO DROP "THETA s" HALT HMS -> 'b'
STO DROP "OFFSET DISTANCE" HALT 'a' STO DROP "NUMBER OR CHORDS" HALT
'd' STO DROP CLLCD 2 FIX "TOTAL CORRECTION" 1 DISP 'pi*b*a/180' EVAL
DUP 'c' STO 3 DISP 4 WAIT "CHORD LENGTH" 1 DISP '2*c/d^2' EVAL 'g'
STO 'l/d+g/2' EVAL DUP 'f' STO 3 DISP 4 WAIT 'd-1' EVAL 'd' STO 1 d
START "CHORD LENGTH" 1 DISP f g + DUP 'f' STO 3 DISP 4 WAIT NEXT { a
b c d f g l } PURGE STD CLMF >>

# ENTER 'SPOFF' STO
```

OFFSET SPIRAL CHORD LENGTH

LENGTH OF SPIRAL	400.00'
THETA Os	7°00'00"
*OFFSET DISTANCE	-30'
NUMBER OF CHORDS	10
TOTAL CORRECTION	-3.67'

CHORD NUMBER	CHORD LENGTH	CHORD NUMBER	CHORD LENGTH
1	39.96	6	39.60
2	39.89	7	39.52
3	39.82	8	39.45
4	39.74	9	39.38
5	39.67	10	39.30

*NOTE: 1) OFFSETS INSIDE THE SPIRAL CURVE ARE NEGATIVE AND OFFSETS OUTSIDE THE CURVE ARE POSITIVE.

2) pi IS KEYED AS ↑π ON THE CALCULATOR.

VERTICAL CURVE ELEVATIONS

```
<< "GRADE 1" HALT 'a' STO DROP "GRADE 2" HALT 'b' STO DROP "LENGTH"
HALT 100 / 'c' STO DROP "VPC STATION" HALT 'd' STO DROP "VPC
ELEVATION" HALT 'g' STO DROP '(b-a)/(2*c)' EVAL 'f' STO { b c }
PURGE STA >>

# ENTER 'VERT' STO


<<"STATION" HALT 'j' STO DROP CLLCD '(j-d)/100' EVAL 'h' STO "FINISH
GRADE" 1 DISP 'h^2*f+a*h+g' EVAL 2 FIX 3 DISP 4 WAIT STD { j h }
PURGE CLMF STA >>

# ENTER 'STA' STO


<< { a b c d f g } PURGE >>

# ENTER 'PURG' STO
```

* NOTE: 1) TO COMPUTE THE DISTANCE TO THE HIGH POINT ON THE VERTICAL CURVE ONCE DATA HAS BEEN INPUT & PROGRAM HAS BEEN HALTED TYPE '(-100*a)/(2*f)' EVAL IN THE SAME DIRECTORY.

2) LIKEWISE FOR THE LOW POINT ON THE VERTICAL CURVE TYPE '(100*a)/(2*f)' EVAL.

TANGENT GRADE

```
<< "ELEVATION 1" HALT 'a' STO DROP "STATION 1" HALT 'b' STO DROP
"ELEVATION 2" HALT 'c' STO DROP "STATION 2" HALT 'd' STO DROP CLLCD
'a-c' EVAL 'f' STO 'b-d' EVAL 'g' STO 'f/g' EVAL 'h' STO "% GRADE" 1
DISP 'h' RCL 100 * 3 DISP 4 WAIT { f g } PURGE CLMF TGR >>

# ENTER 'TANG' STO


<< "STATION" HALT 'd' STO DROP CLLCD 'd-b' EVAL 'c' STO 'h*c' EVAL
'd' STO "TANGENT GRADE" 1 DISP 'd+a' EVAL 2 FIX 3 DISP 4 WAIT STD
CLMF TGR >>

# ENTER 'TGR' STO


<< { a b c d k } PURGE >>

# ENTER 'PURG' STO
```

TANGENT GRADE AND VERTICAL CURVE EXAMPLE

STATION	% GRADE	TANGENT GRADE	FINISH GRADE	3/4" CTS	1 1/2" CBC	SUB GRADE
2+05.6		1895.58	1895.58	95.46	95.26	94.76
2+00	+5.91460 %	1895.25	1895.25	95.13	94.93	94.43
1+90		1894.66	1894.66	94.54	94.34	93.84
1+75		1893.77	1893.90	93.78	93.58	93.08
1+50		1892.29	1893.19	93.07	92.87	92.37
1+40	VC 100 PI FT	1891.70	1893.11	92.99	92.79	92.29
1+25		1892.50	1893.19	93.07	92.87	92.37
1+00	-5.33333 %	1893.83	1893.89	93.77	93.57	93.07
0+90		1894.37	1894.37	94.25	94.05	93.55
0+75		1895.17	1895.17	95.05	94.85	94.35
0+50	VC 50 PI FT	1896.50	1896.35	96.23	96.03	95.53
0+25	-3.00000 %	1897.25	1897.25	97.13	96.93	96.43
0+00		1898.00	1898.00	97.88	97.68	97.18

CALCULATE BEARING USING RIGHT ANGLES

```
<< 0  'a' STO 360 'b' STO 180 'c' STO "END AZIMUTH" HALT HMS-> 'd' STO
DROP "NUMBER OF ANGLES" HALT DUP 'f' STO 'g' STO DROP BRG1 >>

# ENTER 'BRG' STO


<< "BEGINNING AZIMUTH" HALT HMS-> 'h' STO DROP IF g 0 == THEN 'f' RCL 'g'
STO END BRG2 >>

# ENTER 'BRG1' STO


<< "RIGHT ANGLE" HALT HMS-> 'j' STO DROP 'a+j-c+h' EVAL 'k' STO IF k 0 >
THEN BRG3 ELSE k b + 'h' STO NE END >>

# ENTER 'BRG2' STO


<< IF k b > THEN BRG4 ELSE 'k' RCL 'h' STO NE END >>

# ENTER 'BRG3' STO


<< k b - 'h' STO NE >>

# ENTER 'BRG4' STO


<< IF h 90 > THEN SE ELSE CLLCD 'h' RCL ->HMS 5 FIX 1 DISP "NE" 3 DISP 4
WAIT END g 1 - 'g' STO IF g 0 == THEN d h - f / 'a' STO STD CLMF BRG1
ELSE STD CLMF BRG2 END >>

# ENTER 'NE' STO


<< IF h c > THEN SW ELSE CLLCD c h / ->HMS 5 FIX 1 DISP "SE" 3 DISP 4
WAIT END g 1 - 'g' STO IF g 0 == THEN d h - f / 'a' STO STD CLMF BRG1
ELSE STD CLMF BRG2 END >>

# ENTER 'SE' STO


<< IF h 270 > THEN NW ELSE CLLCD h c - ->HMS 5 FIX 1 DISP "SW" 3 DISP 4
WAIT END g 1 - 'g' STO IF g 0 == THEN d h - f / 'a' STO STD CLMF BRG1
ELSE STD CLMF BRG2 END >>

# ENTER 'SW' STO


<<  b h - ->HMS 5 FIX CLLCD 1 DISP "NW" 3 DISP 4 WAIT g 1 - 'g' STO IF g
0 == THEN d h - f / 'a' STO STD CLMF BRG1 ELSE STD CLMF BRG2 END >>

# ENTER 'NW' STO

<< { a b c d f g h j k } >>

# ENTER 'BPUR' STO
```

CALCULATE COORDINATES

```
<< 0 DUP 'l' STO 'n' STO 0 'm' STO "END NORTHING" HALT 'a' STO DROP
"END EASTING" HALT 'c' STO DROP "NUMBER OF LEGS" HALT DUP 'o' STO
'f' STO DROP CORD1 >>

# ENTER 'COORD' STO


<< "BEGINNING NORTHING" HALT 'b' STO DROP "BEGINNING EASTING" HALT
'd' STO DROP CORD2 >>

# ENTER 'CORD1' STO


<< "BEARING ANGLE" HALT HMS-> 'h' STO DROP "QUADRANT" HALT 'k' STO
DROP SUBR >>

# ENTER 'CORD2' STO


<< "DISTANCE" HALT 'g' STO DROP g l + 'l' STO g h R->C P->R DUP RE
'p' STO IM 'j' STO CLLCD "NORTHING" 1 DISP 'g*m+p+b' EVAL DUP 'b'
STO 2 FIX 3 DISP 3 WAIT STD "EASTING" 1 DISP 'g*n+j+d' EVAL DUP 'd'
STO 2 FIX 3 DISP 3 WAIT o 1 - 'o' STO IF o 0 == THEN '(a-b)/l' EVAL
'm' STO '(c-d)/l' EVAL 'n' STO 'f' RCL 'o' STO 0 'l' STO STD CLMF
CORD1 ELSE STD CLMF CORD2 END >>

# ENTER 'CORD3' STO


<< IF k 2 < THEN CORD3 END IF k 3 < THEN SUB1 ELSE SUB2 END >>

# ENTER 'SUBR' STO


<< 180 h - 'h' STO CORD3 >>

# ENTER 'SUB1' STO


<< IF k 3 > THEN SUB3 ELSE h 180 + 'h' STO CORD3 END >>

# ENTER 'SUB2' STO


<< 360 h - 'h' STO CORD3 >>

# ENTER 'SUB3' STO

<< { a b c d f g h j k l m n o p } PURGE >>

# ENTER 'CPUR' STO
```

COORDINATE AND BEARING EXAMPLE

POINT	< #	ANGLE	CALC BEARING	ADJ BEARING	PI TO PI	CALC. CO-ORD NORTHING	CALC. CO-ORD EASTING	ADJ. CO-ORD NORTHING	ADJ. CO-ORD EASTING
I-2 △				S04°10'53.9"W					
602 △	1	58°38'20"	N62°49'13.9"E			5702.72	31684.94	5702.72	31684.94
				N62°49'25.9"E	1281.63'				
I-2A	2	111°50'06.7"	N05°20'39.4"W			6288.08	32825.09	6288.08	32825.12
				N05°20'15.4"W	962.91'				
I-2B	3	183°53'43.3"	N01°26'56.1"W			7246.81	32735.51	7246.82	32735.58
				N01°26'20.2"W	1407.87'				
I-2C	4	183°11'23.3"	N01°44'27.2"E			8654.24	32700.16	8654.25	32700.26
				N01°45'15.1"E	2716.49'				
603 △		168°10'13.3"				11369.45	32783.32	11369.48	32783.50
				N10°04'19.6"W					
I-3 △								12604.44	32564.14
				CLOSURE = 1:35,000				:0.03	:0.18

SIDESHOTS FOR COORDINATE AND BEARING

```
<< "BACK NORTHING" HALT 'a' STO DROP "BACK EASTING" HALT 'b' STO DROP
"INSTR. NORTHING" HALT 'c' STO DROP "INSTR. EASTING" HALT 'd' STO DROP
180 'h' STO c a - d b - R->C R->P IM DUP 'j' STO 'l' STO SIDE1 >>

# ENTER 'SSHOT' STO


<< "ANGLE RIGHT" HALT HMS-> 'f' STO DROP "DISTANCE" HALT 'g' STO DROP
'f+l-h' EVAL 'j' STO IF j 0 < THEN 2 h * j + 'j' STO SUBR END IF j 2 h
* > THEN 'j-2*h' EVAL 'j' STO SUBR ELSE SUBR END >>

# ENTER 'SIDE1' STO


<< "NORTHING COORD" 1 DISP g j R->C P->R DUP RE c + 2 FIX 3 DISP 3
WAIT STD "EASTING COORD" 1 DISP IM d + 2 FIX 3 DISP 3 WAIT STD CLMF
SIDE1 >>

# ENTER 'SIDE2'


<< IF j 90 > THEN SUB1 ELSE CLLCD "BEARING ANGLE" 1 DISP 'j' RCL ->HMS
5 FIX 3 DISP "NE" 4 DISP 3 WAIT STD CLMF SIDE2 END >>

# ENTER 'SUBR' STO


<< IF j h > THEN SUB2 ELSE CLLCD "BEARING ANGLE" 1 DISP h j - ->HMS 5
FIX 3 DISP "SE" 4 DISP 3 WAIT STD CLMF SIDE2 END >>

# ENTER 'SUB1' STO


<< IF j 270 > THEN SUB3 ELSE CLLCD "BEARING ANGLE" 1 DISP j h -
->HMS 5 FIX 3 DISP "SW" 4 DISP 3 WAIT STD CLMF SIDE2 END >>

# ENTER 'SUB2' STO


<< CLLCD "BEARING ANGLE" 1 DISP h 2 * j - ->HMS 5 FIX 3 DISP "NW" 4
DISP 3 WAIT STD CLMF SIDE2 >>

# ENTER 'SUB3' STO


<< { a b c d f g h j k l } PURGE >>

# ENTER 'SSPUR' STO
```

COORDINATE SIDESHOT EXAMPLE

	- C O O R D I N A T E S -	
	NORTHING	EASTING
BACKSITE STATION	5000.00	35,000.00
INSTRUMENT STATION	4175.10	34,276.86

POINT #	ANGLE RIGHT	DISTANCE	BEARING	NORTHING	EASTING
#1	345°09'27"	264.00'	N26 23'47.7"E	4411.57	34,394.23
#2	120°18'27"	1618.00'	S18 27'12.3"E	2640.30	34,789.01
#3	187°25'15"	641.73'	S48 39'35.7"W	3751.22	33,795.05
#4	273°15'30"	516.74'	N45 30'09.3"W	4537.27	33,908.28

INVERSE BETWEEN COORDINATES

```
<< "NORTHING 1" HALT 'a' STO DROP "EASTING 1" HALT 'b' STO DROP
"NORTHING 2" HALT 'c' STO DROP "EASTING 2" HALT 'd' STO DROP c a - d
b - R->C R->P DUP RE 'k' STO IM 'j' STO CLLCD "DISTANCE" 1 DISP 'k'
RCL 2 FIX 3 DISP 3 WAIT STD IF j 0 < THEN j 360 + 'j' STO NE ELSE NE
END >>

# ENTER 'INVER' STO


<< IF j 90 > THEN SE ELSE "BEARING ANGLE" 1 DISP 'j' RCL ->HMS 5 FIX
3 DISP "NE" 4 DISP 3 WAIT STD CLMF INVER END >>

# ENTER 'NE' STO


<< IF j 180 > THEN SW ELSE "BEARING ANGLE" 1 DISP 180 j - ->HMS 5
FIX 3 DISP "SE" 4 DISP 3 WAIT STD CLMF INVER END >>

# ENTER 'SE' STO


<< IF j 270 > THEN NW ELSE "BEARING ANGLE" 1 DISP j 180 - ->HMS 5
FIX 3 DISP "SW" 4 DISP 3 WAIT STD CLMF INVER END >>

# ENTER 'SW' STO


<< "BEARING ANGLE" 1 DISP 360 j - ->HMS 5 FIX 3 DISP "NW" 4 DISP 3
WAIT STD CLMF INVER >>

# ENTER 'NW' STO


<< { a b c d j k } PURGE >>

# ENTER 'INPUR' STO
```

INVERSE BETWEEN COORDINATES EXAMPLE

POINT #	NORTHING	EASTING	DISTANCE	BEARING
602	5702.72	31,684.94		
I-2	8942.36	31,921.80	3248.29'	N04° 10'53.9"E
I-1A	4175.10	34,276.87		
I-1B	2640.28	34,789.03	1618.02'	S18° 27'12.7"E
601	5000.00	35,000.00		
I-1A	4175.10	34,276.87	1096.99'	S41° 14'19.3"W
I-1D	4554.62	34,085.94		
I-1E	5361.18	33,561.01	962.34'	N33° 03'25.6"W

TOPOG FROM POINT ON TANGENT WITH E.D.M.

```
<<  "INSTRUMENT STATION" HALT 's' STO DROP POT2 >>

# ENTER 'POT' STO


<<  "ANGLE" HALT HMS -> 'a' STO DROP "DISTANCE" HALT 'd' STO DROP 2
FIX DEG d a R->C P->R IM 'C' STO CLLCD IF C 0 < THEN 'c' SNEG
"STATION" 1 DISP d a R->C P->R RE 'b' STO 'b+s' EVAL 3 DISP 4 WAIT
"CL REFERENCE" 1 DISP 'c' RCL 3 DISP "LEFT" 4 DISP 4 WAIT { a b c d
} PURGE STD CLMF POT2 ELSE d a R->C P->R RE 'b' STO "STATION" 1 DISP
'b+s' EVAL 3 DISP 4 WAIT d a R->C P->R IM 'c' STO "CL REFERENCE" 1
DISP 'c' RCL 3 DISP "RIGHT" 4 DISP 4 WAIT { a b c d } PURGE STD CLMF
POT2 END >>

# ENTER 'POT2' STO
```

EXAMPLE

	INSTRUMENT STATION	26+64.7
1)	ANGLE RIGHT	60°15'40"
	DISTANCE	197.63'
	OBJECT 1 IS 171.60' RIGHT OF STATION	27+62.73
2)	ANGLE RIGHT	240°43'20"
	DISTANCE	264.82'
	OBJECT 2 IS 230.99' LEFT OF STATION	25+35.19

* NOTE: BACK SITE ALONG THE BACK TANGET WITH INSTRUMENT BEFORE TURNING ANGLES AND SHOOTING DISTANCES.

TOPOG FROM POINT OF CURVE WITH E.D.M.

```
<<    "INSTRUMENT STATION" HALT 's' STO DROP "DEGREE OF CURVE" HALT
HMS -> 'd' STO DROP "CURVE RT=0 LT=1" HALT 'z' STO DROP '5728.578/d'
EVAL 'r' STO POC1 >>

# ENTER 'POC' STO


<< "ANGLE" HALT HMS -> 'a' STO DROP "DISTANCE" HALT 'b' STO DROP DEG
b a R->C P->R DUP IM 'x' STO RE 'y' STO IF z 0 == THEN 'x' SNEG END
'r+x' EVAL 'f' STO y f R->C R->P IM 'o' STO '(90-o)/d*100+s' EVAL
'h' STO CLLCD "STATION" 1 DISP 'h' RCL 3 DISP 4 WAIT IF z 1 == THEN
CRLT ELSE y f R->C R->P RE 'l' STO 'l-r' EVAL 'c' STO IF C 0 < THEN
'c' SNEG "CL REFERENCE" 1 DISP 'c' RCL 3 DISP "RIGHT" 4 DISP 4 WAIT
ELSE "CL REFERENCE" 1 DISP 'c' RCL 3 DISP "LEFT" 4 DISP 4 WAIT END {
a b c f h l o x y } PURGE STD CLMF POC1 END >>

# ENTER 'POC1' STO


<< y f R->C r->P RE 'l' STO 'r-l' EVAL 'c' STO IF C 0 < THEN 'c'
SNEG "CL REFERENCE" 1 DISP 'c' RCL 3 DISP "RIGHT" 4 DISP 4 WAIT ELSE
"CL REFERENCE" 1 DISP 'c' RCL 3 DISP "LEFT" 4 DISP 4 WAIT END { a b
c f h l o c y } PURGE STD CLMF POC1 >>

# ENTER 'CRLT' STO
```

EXAMPLE

INSTRUMENT STATION	208+50
Dc	06°00'00"
ANGLE RIGHT	79°23'40"
DISTANCE	290.78'

1) IF CURVE IS RIGHT THEN:

   STATION = 209+26.20
   CENTER LINE REFERENCE = 283.7' RIGHT

2) IF CURVE IS LEFT THEN:

   STATION = 208+91.20
   CENTER LINE REFERENCE = 287.0' RIGHT

* NOTE: BACK SITE THE P.C. WITH INSTRUMENT BEFORE TURNING ANGLES AND SHOOTING DISTANCES.

MEAN ZENITH ANGLE AND COMPUTE HORIZONTAL DISTANCE

```
<< "DIR ZENITH ANGLE" HALT HMS -> 'a' STO DROP "REVERSE ZENITH
ANGLE" HALT HMS -> 'r' STO DROP "SLOPE DISTANCE" HALT 'b' STO DROP
CLLCD '.5*(360-t+a)' EVAL 'a' STO "MEAN ANGLE" 1 DISP 'a' RCL -> HMS
5 FIX 3 DISP 4 WAIT "HORIZONTAL DISTANCE" 1 DISP 'SIN(a)*b' EVAL 3
FIX 3 DISP 4 WAIT { a r b } PURGE STD CLMF >>

# ENTER 'MNZE' STO
```

EXAMPLE

DIRECT ZENITH ANGLE	88° 34'33"
REVERSE ZENITH ANGLE	271° 25'45"
SLOPE DISTANCE	342.86'
MEAN ANGLE	88° 34'24"
HORIZONTAL DISTACE	342.75'