9/4.4.6

RSX SEC : sécante
RSX ASEC : arc sécante

RSX SEC : SÉCANTE

La fonction sécante ne fait pas partie de la bibliothèque standard des CPC. Il est cependant possible de la simuler en utilisant une formule équivalente composée de fonctions qui font partie de la bibliothèque standard des CPC. En effet :

SEC(X) = 1 / COS(X)

Comment exécuter le programme

Le programme mettant en œuvre une RSX, il est forcément écrit en Assembleur. Si vous désirez l'utiliser sous sa forme Assembleur, entrez le listing des pages suivantes :

```
1                               ORG  9000H
2                               LOAD 9000H
3                       ;-------------------------------
4                       ; RSX SECANTE
5                       ; Format : |SEC,@VAR
6                       ; Entree (VAR)=Angle
7                       ; Sortie (VAR)=SECANTE de (VAR)
8                       ;-------------------------------
9                       ;
10                      ;
11                      ;---------------------------------
12                      ; Declaration des constantes
13                      ; et des variables du programme
14                      ;---------------------------------
15                      ;
16                      DEG:        EQU  0B113H         ;Unite expr angles
17                      ENTFLO:     EQU  0BD61H         ;Ent->Flottant
18                      DIVFLO:     EQU  0BD85H         ;Division flot
19                      COS:        EQU  0BDACH         ;Cosinus
20                      SAVHL:      DS   2              ;Sauvegarde de HL
21                      Z1:         DS   5              ;Zone reel 1
22                      Z3:         DS   5              ;Zone reel 3
23                      LOGEXT:     EQU  0BCD1H         ;KL LOG EXT
24                      BUF:        DS   4              ;Zone RAM pour LOG EXT
25 9010 1590            PTRTAB:     DW   TABLE          ;Pointeur TABLE
26 9012 C32390                      JP   SEC            ;Traitement du SEC
27 9015 5345            TABLE:      DB   "SE"
28 9017 C3                          DB   "C"+80H
29 9018 00                          DB   0              ;Fin de table
30                      ;
31                      ;---------------------------------
```

```
                        ; Definition de la RSX
                        ;---------------------------------
                        ;
                        DEFRSX:     EQU  $              ;Point d'entree
9019 011090             DEFRSX:     LD   BC,PTRTAB      ;Ptr table definition
901C 210C90             DEFRSX:     LD   HL,BUF         ;Buffer pour LOG EXT
901F CDD1BC             DEFRSX:     CALL LOGEXT         ;Definition de la RSX
9022 C9                 DEFRSX:     RET
                        ;
                        ;---------------------------------
                        ; Traitement de SEC
                        ;---------------------------------
                        ;
                        SEC:        EQU  $              ;Point d'entree
9023 DD6601             SEC:        LD   H,(IX+1)
9026 DD6E00             SEC:        LD   L,(IX+0)       ;Adresse de la var.
9029 220090             SEC:        LD   (SAVHL),HL     ;Sauvegarde
902C CD5790             SEC:        CALL ZONE1          ;Memorisation
                        ;
902F 3EFF               SEC:        LD   A,0FFH
9031 3213B1             SEC:        LD   (DEG),A        ;Unite d'angle=Degre
                        ;
9034 210290             SEC:        LD   HL,Z1
9037 CDACBD             SEC:        CALL COS            ;COS(X)
903A 210100             SEC:        LD   HL,1
903D 110790             SEC:        LD   DE,Z3
9040 CD61BD             SEC:        CALL ENTFLO         ;1 en flottant
9043 210790             SEC:        LD   HL,Z3
9046 110290             SEC:        LD   DE,Z1
```

```
61 9049 CD85BD              CALL DIVFLO          ;1/COS(X)
62              ;
63              FINPRE:     EQU  $
64 904C 210790              LD   HL,Z3
65 904F ED5B0090            LD   DE,(SAVHL)
66 9053 CD6090              CALL FLODEHL         ;Resultat
67              ;
68              ;
69              FIN:        EQU  $               ;Fin du programme
70 9056 C9                  RET
71              ;
72              ;---------------------------------
73              ; Zone des sous-programmes
74              ;---------------------------------
75              ;
76              ;---------------------------------
77              ; Transfert des BC octets pointes
78              ; par HL dans le buffer Z1
79              ;---------------------------------
80              ;
81              ZONE1:      EQU  $
82 9057 110290              LD   DE,Z1
83 905A 010500              LD   BC,5
84 905D EDB0                LDIR
85 905F C9                  RET
86              ;
87              ;---------------------------------
88              ; Transfert flottant de (HL)
89              ; dans (DE)
90              ;---------------------------------
91              :
```

```
92                  FLODEHL:    EQU  $
93 9060 010500                  LD   BC,5
94 9063 EDB0                    LDIR
95 9065 C9                      RET
96                              END
```

ADDFLO	BD79	ATAN	BDB2	ACOT	9024	BUF	900C
DEG	B113	DEFRSX	901A	ENTFLO	BD61	FINPRE	9054
FIN	905E	FLODEHL	9068	LOGEXT	BCD1	NEGFLO	BD8E
PTRTAB	9010	SAVHL	9000	TABLE	9015	Z1	9002
Z3	9007	ZONE1	905F				

Installez la RSX en tapant sous Basic :

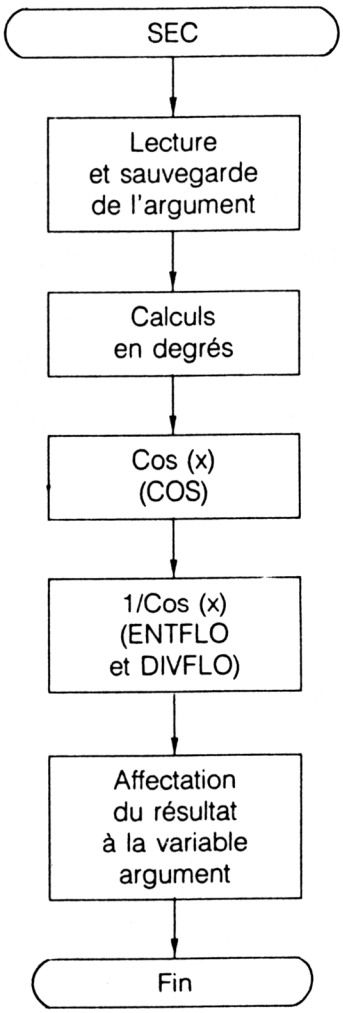
CALL &9019

La fonction **¦SEC** fait maintenant partie du Basic standard. Voici comment l'utiliser :

```
10 a=.5        'Argument à passer à la fonction SEC
20 ¦SEC,@a     'Appel de la fonction SEC
30 PRINT a     'Affichage du résultat
```

Le programme en détail

La logique du programme obéit à l'ordinogramme suivant :



Les premières lignes du programme laissent apparaître des déclarations de constantes et variables.

Signalons en particulier :

```
DEG:      EQU    0B113H      ; Unité d'expression des angles
ENTFLO:   EQU    0BD61H      ; Conversion Entier → Flottant
DIVFLO:   EQU    0BD85H      ; Division de deux flottants
COS:      EQU    0BDACH      ; Cosinus
```

Ces adresses sont correctes pour les CPC 664. Si vous possédez un CPC 464 ou un CPC 6128, vous devez les convertir comme suit :

Point d'entrée	CPC 464	CPC 664	CPC 6128
ENTFLO	0BD40H	0BD61H	0BD64H
DIVFLO	0BD64H	0BD85H	0BD88H
DEG	0B8F7H	0B113H	0B113H
COS	0BD8BH	0BDACH	0BDAFH

Remarque :

Pour faciliter l'utilisation de cette RSX, les chargeurs Basic sont donnés dans les trois versions (464, 664 et 6128).

Les zones Z1 et Z3 de cinq octets sont utilisées pour manipuler les nombres flottants.

La macro **LOGEXT** permet de déclarer la nouvelle RSX.

Les structures **BUF, PTRTAB** et **TABLE** sont typiques des RSX :

— **BUF** est un buffer utilisé en interne par la RX ;

— **PTRTAB** contient un pointeur sur la structure **TABLE** et un débranchement au programme de traitement de la RSX ;

— **TABLE** contient le nom de la RSX. La dernière lettre de ce nom doit être masquée pour que son bit de poids fort soit à un. Enfin, le nom de la dernière RSX doit être suivi d'un octet nul. Dans notre cas, la RSX **SEC** étant la seule utilisée, cet octet nul se trouve immédiatement après le nom de la RSX.

La définition de la RSX doit être effectuée avant sa première utilisation. Le court programme situé à l'étiquette **DEFRSX** est chargé de cette tâche.

Ce programme fait appel à la macro **LOGEXT** en lui transmettant :

— l'adresse de la table de définition dans **BC**,

— l'adresse d'un buffer interne **BUF**.

La logique de ce programme est la même que celle des RSX définies dans le complément 18. Reportez-vous à la RSX **ACOS** pour tout renseignement.

Lorsque l'utilisateur désire activer la fonction SEC, il place l'argument de la fonction dans une variable à la RSX **|SEC**. Le résultat est retourné dans cette même variable.

Par exemple, pour connaître la sécante de 30, tapez :

```
b = 30: |SEC,@b:? b
```

L'ordinateur affichera 1.1547005.

Lorsque l'interpréteur rencontre le caractère | (code ASCII 124), il se reporte à la table des RSX à la recherche de la fonction spécifiée. Si la fonction

existe dans cette table, elle est exécutée. Dans le cas contraire, un message d'erreur est affiché sur l'écran.

Donc, si vous tapez ¦ SEC, l'interpréteur recherchera SEC dans la table RSX. L'adresse de traitement SEC lui étant affectée, la routine située à l'étiquette SEC sera exécutée.

La première action effectuée dans cette routine consiste à récupérer l'adresse de la variable passée à l'aide du registre IX. Cette adresse est sauvegardée dans la variable SAVHL, et la valeur réelle qu'elle contient dans la zone réelle Z1 :

```
SEC:    EQU     $
        LD      H,(IX+1)
        LD      L,(IX+0)        ; Adresse de la variable
        LD      (SAVHL),HL      ; Sauvegarde adresse
        CALL    ZONE1           ; Mémorisation valeur
```

Les angles passés à la RSX étant toujours exprimés en degrés, les lignes suivantes initialisent la variable d'expression des angles :

```
        LD      A,0FFH
        LD      (DEG),A         ; Unité d'angle = degré
```

Le cosinus de l'angle passé est calculé à l'aide de la fonction COS :

```
        LD      HL,Z1
        CALL    COS             ; COS(X)
```

Pour pouvoir calculer la quantité 1/COS(X), il faut d'abord convertir la valeur 1 en réel :

```
        LD      HL,1
        LD      DE,Z3
        CALL    ENTFLO          ; 1 en flottant
```

La sécante de l'angle passé peut maintenant être calculée :

```
        LD      HL,Z3
        LD      DE,Z1
        CALL    DIVFLO          ; 1/COS(X)
```

Cette quantité est affectée à la variable passée en entrée à l'aide des instructions suivantes :

```
FINPRE: EQU     $
        LD      HL,Z3
        LD      DE,(SAVHL)
        CALL    FLODEHL         ; Résultat
```

Le sous-programme ZONE1 utilise l'instruction LDIR pour transférer les 5 octets pointés par HL dans le buffer Z1 :

```
ZONE1:  EQU     $
        LD      BC,5
        LDIR
```

Le sous-programme FLODEHL transfère les 5 octets pointés par HL à partir de l'adresse DE :

```
FLODEHL: EQU    $
        LD      BC,5
        LDIR
```

Si vous préférez utiliser un chargeur Basic, voici le listing et les données de checksum correspondantes :

```
1000 '------------------------------
1010 ' Chargeur de la RSX SECANTE
1020 '------------------------------
1030 ' VERSION CPC 464
1040 '------------------------------
1050 '
1060 FOR I=&9000 TO &90ED
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '------------------------------
1130 ' Codes op de la RSX SECANTE
1140 '------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 15,90,C3,23,90,53,45,C3,0,1,10,90,21,C,90,CD
1180 DATA D1,BC,C9,DD,66,1,DD,6E,0,22,0,90,CD,57,90,3E
1190 DATA FF,32,F7,B8,21,2,90,CD,8B,BD,21,1,0,11,7,90
1200 DATA CD,40,BD,21,7,90,11,2,90,CD,64,BD,21,7,90,ED
1210 DATA 5B,0,90,CD,60,90,C9,11,2,90,1,5,0,ED,B0,C9
1220 DATA 1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0
```

```
0 A6 90 78 BE 86 6E
```

```
1000 '------------------------------
1010 ' Chargeur de la RSX SECANTE
1020 '------------------------------
1030 ' VERSION CPC 664
1040 '------------------------------
1050 '
1060 FOR I=&9000 TO &9065
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '------------------------------
1130 ' Codes op de la RSX SECANTE
1140 '------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 15,90,C3,23,90,53,45,C3,0,1,10,90,21,C,90,CD
1180 DATA D1,BC,C9,DD,66,1,DD,6E,0,22,0,90,CD,57,90,3E
1190 DATA FF,32,13,B1,21,2,90,CD,AC,BD,21,1,0,11,7,90
1200 DATA CD,61,BD,21,7,90,11,2,90,CD,85,BD,21,7,90,ED
1210 DATA 5B,0,90,CD,60,90,C9,11,2,90,1,5,0,ED,B0,C9
1220 DATA 1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0
```

```
0 A6 90 AD 1 B6 6E
```

```
1000 '------------------------------
1010 ' Chargeur de la RSX SECANTE
1020 '------------------------------
1030 ' VERSION CPC 6128
1040 '------------------------------
1050 '
1060 FOR I=&9000 TO &90ED
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '------------------------------
1130 ' Codes op de la RSX SECANTE
1140 '------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 15,90,C3,23,90,53,45,C3,0,1,10,90,21,C,90,CD
1180 DATA D1,BC,C9,DD,66,1,DD,6E,0,22,0,90,CD,57,90,3E
1190 DATA FF,32,13,B1,21,2,90,CD,AF,BD,21,1,0,11,7,90
1200 DATA CD,64,BD,21,7,90,11,2,90,CD,88,BD,21,7,90,ED
1210 DATA 5B,0,90,CD,60,90,C9,11,2,90,1,5,0,ED,B0,C9
1220 DATA 1,5,0,ED,B0,C9,0,0,0,0,0,0,0,0,0,0
```

```
0 A6 90 B0 7 86 6E
```

RSX ASEC : ARC SÉCANTE

La fonction arc sécante ne fait pas partie de la bibliothèque standard des CPC. Pour la simuler, nous allons faire appel à la formule suivante :

```
ASEC(X) = ATN(RAC(X^2 - 1)) + (SGN(X) - 1) * 90
```

Comment exécuter le programme

Si vous désirez utiliser le programme sous sa forme Assembleur, entrez le listing suivant :

```
1                       ORG  9000H
2                       LOAD 9000H
3               ;------------------------------
4               ; RSX ARC SECANTE
5               ; Format : !ASEC,@VAR
6               ; Entree (VAR)=Angle
7               ; Sortie (VAR)=ARC SEC de (VAR)
8               ;------------------------------
9               ;
10              ;
11              ;--------------------------------
12              ; Declaration des constantes
13              ; et des variables du programme
14              ;--------------------------------
15              ;
16              DEG:        EQU  0B113H        ;Unite expr angles
17              ENTFLO:     EQU  0BD61H        ;Ent->Flottant
18              ADDFLO:     EQU  0BD79H        ;Addtion flot
19              DIVFLO:     EQU  0BD85H        ;Division flot
20              RACFLO:     EQU  0BD9AH        ;Racine carree
21              MULFLO:     EQU  0BD82H        ;Multiplication
22              NEGFLO:     EQU  0BD8EH        ;Negation flot
23              ATAN:       EQU  0BDB2H        ;Arc tangente
24              SIG:        EQU  0BD91H        ;Signe d'1 flottant
25              SAVHL:      DS   2             ;Sauvegarde de HL
26              Z1:         DS   5             ;Zone reel 1
27              Z2:         DS   5             ;Zone reel 2
28              Z3:         DS   5             ;Zone reel 3
29              Z4:         DS   5             ;Zone reel 4
30              LOGEXT:     EQU  0BCD1H        ;KL LOG EXT
31              BUF:        DS   4             :Zone RAM pour LOG EXT
```

```
32 901A 1F90     PTRTAB:     DW   TABLE            ;Pointeur TABLE
33 901C C32E90               JP   ASEC             ;Traitement du ASEC
34 901F 415345   TABLE:      DB   "ASE"
35 9022 C3                   DB   "C"+80H
36 9023 00                   DB   0                ;Fin de table
37               ;
38               ;--------------------------------
39               ; Definition de la RSX
40               ;--------------------------------
41               ;
42               DEFRSX:     EQU  $                ;Point d'entree
43 9024 011A90               LD   BC,PTRTAB        ;Ptr table definition
44 9027 211690               LD   HL,BUF           ;Buffer pour LOG EXT
45 902A CDD1BC               CALL LOGEXT           ;Definition de la RSX
46 902D C9                   RET
47               ;
48               ;--------------------------------
49               ; Traitement de ASEC
50               ;--------------------------------
51               ;
52               ASEC:       EQU  $                ;Point d'entree
53 902E DD6601               LD   H,(IX+1)
54 9031 DD6E00               LD   L,(IX+0)         ;Adresse de la var.
55 9034 220090               LD   (SAVHL),HL       ;Sauvegarde
56 9037 CDD690               CALL ZONE1            ;Memorisation
57 903A 2A0090               LD   HL,(SAVHL)
58 903D CDDF90               CALL ZONE3            ;Memorisation
59               ;
60 9040 3EFF                 LD   A,0FFH
```

```
61 9042 3213B1                LD    (DEG),A          ;Unite d'angle=Degr@
62                    ;
63 9045 210290                LD    HL,Z1
64 9048 110C90                LD    DE,Z3
65 904B CD82BD                CALL  MULFLO           ;X*X
66 904E AF                    XOR   A
67 904F 210100                LD    HL,1
68 9052 110790                LD    DE,Z2
69 9055 CD61BD                CALL  ENTFLO           ;1 en flottant
70 9058 210790                LD    HL,Z2
71 905B CD8EBD                CALL  NEGFLO           ;-1 en flottant
72 905E 210290                LD    HL,Z1
73 9061 110790                LD    DE,Z2
74 9064 CD79BD                CALL  ADDFLO           ;X*X - 1
75 9067 210290                LD    HL,Z1
76 906A CD9ABD                CALL  RACFLO           ;RAC(X*X-1)
77 906D 210290                LD    HL,Z1
78 9070 CDB2BD                CALL  ATAN             ;ATAN(RAC(X*X-1))
79 9073 210C90                LD    HL,Z3
80 9076 CD91BD                CALL  SIG              ;Signe de X
81 9079 FEFF                  CP    255              ;Negatif ? '
82 907B 280C                  JR    Z,NBRENEG        ;Traitement corresp
83
84 907D 2600                  LD    H,0
85 907F 6F                    LD    L,A
86 9080 110790                LD    DE,Z2
87 9083 AF                    XOR   A
88 9084 CD61BD                CALL  ENTFLO           ;0 ou 1 en flottant
89 9087 1810                  JR    SUITE
90                NBRENEG:    EQU   $
91 9089 AF                    XOR   A
```

```
 92 908A 210100              LD   HL,1
 93 908D 110790              LD   DE,Z2
 94 9090 CD61BD              CALL ENTFLO          ;1 en flottant
 95 9093 210790              LD   HL,Z2
 96 9096 CD8EBD              CALL NEGFLO          ;-1 en flottant
 97              SUITE:      EQU  $
 98 9099 AF                  XOR  A
 99 909A 210100              LD   HL,1
100 909D 111190              LD   DE,Z4
101 90A0 CD61BD              CALL ENTFLO          ;1 en flottant
102 90A3 211190              LD   HL,Z4
103 90A6 CD8EBD              CALL NEGFLO          ;-1 en flottant
104 90A9 210790              LD   HL,Z2
105 90AC 111190              LD   DE,Z4
106 90AF CD79BD              CALL ADDFLO          ;SGN(X)-1
107 90B2 AF                  XOR  A
108 90B3 215A00              LD   HL,90
109 90B6 111190              LD   DE,Z4
110 90B9 CD61BD              CALL ENTFLO          ;90 en flottant
111 90BC 210790              LD   HL,Z2
112 90BF 111190              LD   DE,Z4
113 90C2 CD82BD              CALL MULFLO          ;(SGN(X)-1)*90
114 90C5 110290              LD   DE,Z1
115 90C8 CD79BD              CALL ADDFLO          ;ATAN(RAC(X*X-1))+
116             ;(SGN(X)-1)*90
117             ;
118             FINPRE:      EQU  $
119 90CB 210790              LD   HL,Z2
120 90CE ED5B0090            LD   DE,(SAVHL)
```

```
121 90D2 CDE890             CALL FLODEHL          ;Resultat
122              ;
123              ;
124              FIN:       EQU  $                ;Fin du programme
125 90D5 C9                 RET
126              ;
127              ;---------------------------------
128              ; Zone des sous-programmes
129              ;---------------------------------
130              ;
131              ;---------------------------------
132              ; Transfert des BC octets pointes
133              ; par HL dans le buffer Z1
134              ;---------------------------------
135              ;
136              ZONE1:     EQU  $
137 90D6 110290             LD   DE,Z1
138 90D9 010500             LD   BC,5
139 90DC EDB0               LDIR
140 90DE C9                 RET
141              ;
142              ;---------------------------------
143              ; Transfert des BC octets pointes
144              ; par HL dans le buffer Z3
145              ;---------------------------------
146              ;
147              ZONE3:     EQU  $
148 90DF 110C90             LD   DE,Z3
149 90E2 010500             LD   BC,5
150 90E5 EDB0               LDIR
151 90E7 C9                 RET
```

```
152                     ;
153                     ;--------------------------------
154                     ; Transfert flottant de (HL)
155                     ; dans (DE)
156                     ;--------------------------------
157                     ;
158                     FLODEHL:    EQU  $
159 90E8 010500                    LD   BC,5
160 90EB EDB0                       LDIR
161 90ED C9                         RET
162                                 END
```

ADDFLO	BD79	ATAN	BDB2	ACOT	9024	BUF	900C
DEG	B113	DEFRSX	901A	ENTFLO	BD61	FINPRE	9054
FIN	905E	FLODEHL	9068	LOGEXT	BCD1	NEGFLO	BD8E
PTRTAB	9010	SAVHL	9000	TABLE	9015	Z1	9002
Z3	9007	ZONE1	905F				

Installez la RSX en tapant sous Basic :

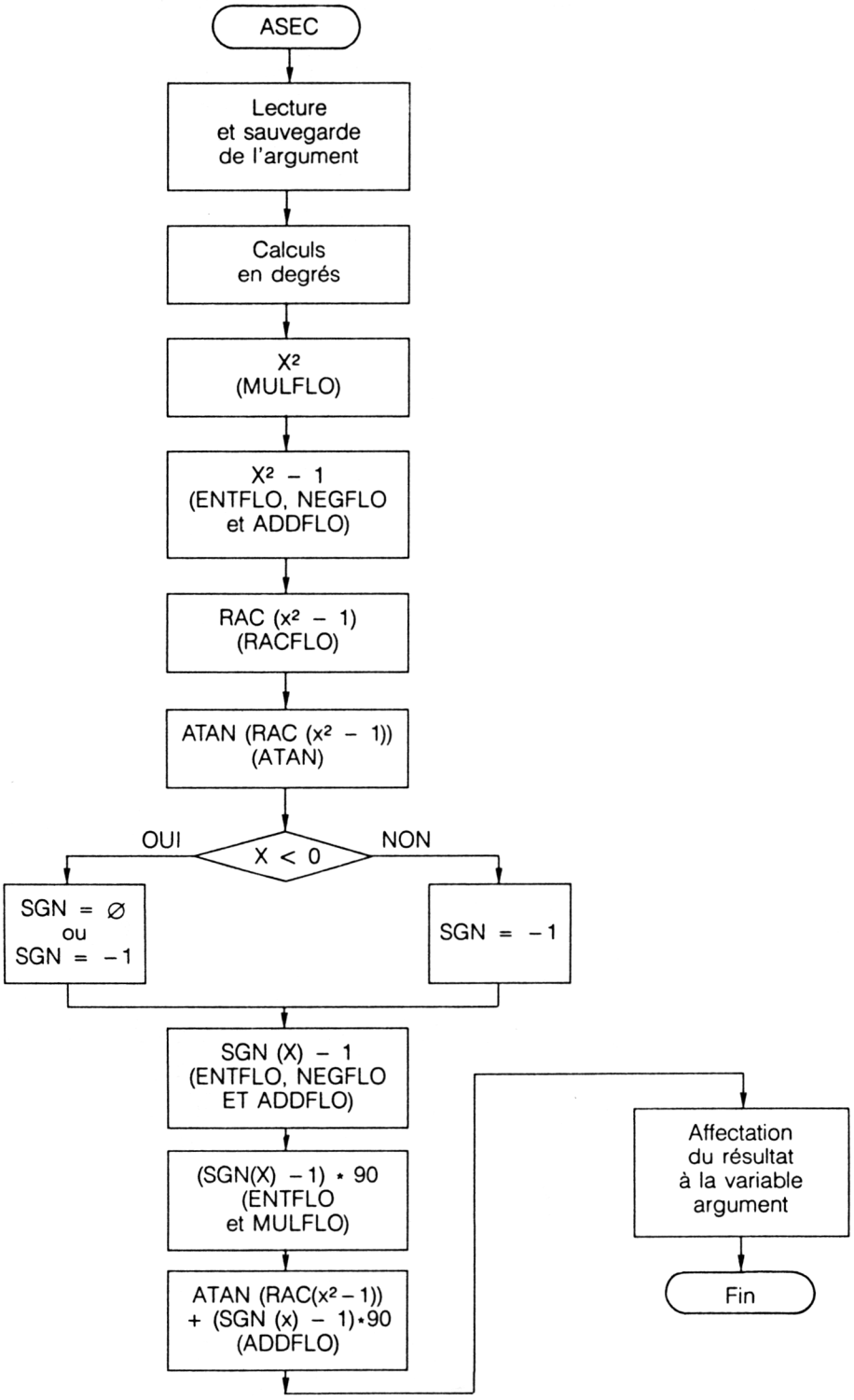
CALL &9024

La fonction **¦ASEC** fait maintenant partie du Basic standard. Voici comment l'utiliser :

```
10 a = 1.1547005        'Argument à passer à la fonction ASEC
20 ¦ASEC,@a             'Appel de la fonction ASEC
30 PRINT a              'Affichage du résultat
```

LE PROGRAMME EN DÉTAIL

La logique du programme obéit à l'ordinogramme suivant :



Le programme ayant été développé sur un CPC 664, il est nécessaire de modifier l'adresse de tous les vecteurs mathématiques utilisés si vous possédez un 464 ou un 6128 :

Point d'entrée	CPC 464	CPC 664	CPC 6128
ENTFLO	0BD40H	0BD61H	0BD64H
ADDFLO	0BD58H	0BD79H	0BD7CH
DIVFLO	0BD64H	0BD85H	0BD88H
RACFLO	0BD79H	0BD9AH	0BD9DH
MULFLO	0BD61H	0BD82H	0BD85H
NEGFLO	0BD6DH	0BD8EH	0BD91H
ATAN	0BD91H	0BDB2H	0BDB5H
SIG	0BD70H	0BD91H	0BD94H
DEG	0B8F7H	0B113H	0B113H

Pour faciliter l'utilisation de cette RSX, les chargeurs Basic sont donnés dans les trois versions (464, 664 et 6128).

La définition de la RSX doit être effectuée avant sa première utilisation. Le court programme situé à l'étiquette **DEFRSX** est chargé de cette tâche.

Ce programme est identique à celui des fonctions précédentes.

Lorsque l'utilisateur désire utiliser la fonction **ASEC**, il place l'argument de la fonction dans une variable flottante, et passe l'adresse de cette variable à la RSX **¦ASEC**. Le résultat est retourné dans cette même variable.

Par exemple, pour connaître l'arc sécante de 1.1547005, tapez :

```
b = 1.1547005: ¦ ASEC,@b:? b
```

L'ordinateur affichera 30.

Lorsque l'interpréteur rencontre l'instruction **¦ASEC**, il recherche **ASEC** dans la table RSX. L'adresse de traitement ASEC lui étant affectée, la routine située à l'étiquette **ASEC** est exécutée.

La première action effectuée dans cette routine consiste à récupérer l'adresse de la variable passée à l'aide du registre **IX**. Cette adresse est sauvegardée dans la variable **SAVHL**, et la valeur réelle qu'elle contient dans les zones réelles Z1 et Z3 :

```
ASEC: EQU   $
      LD    H, (IX + 1)
      LD    L, (IX + 0)      ;Adresse de la variable
      LD    (SAVHL),HL       ;Sauvegarde adresse
      CALL  ZONE1            ;Mémorisation valeur
      LD    (SAVHL),HL       ;Sauvegarde adresse
      CALL  ZONE3            ;Mémorisation valeur
```

Les angles passés à la RSX étant toujours exprimés en degrés, les lignes suivantes initialisent la variable d'expression des angles :

```
      LD    A.0FFH
      LD    (DEG),A          ;Unité d'angle = degré
```

Rappelons que l'arc sécante est calculée à l'aide de la formule suivante :

ASEC(X) = ATN(RAC(X^2 – 1)) + (SGN(X) – 1) * 90

La quantité X^2 est calculée à l'aide de la fonction MULFLO :

```
LD    HL,Z1
LD    DE,Z3
CALL  MULFLO    ;X^2
```

La soustraction de la valeur 1 n'est pas aussi simple que l'on pourrait s'y attendre. Dans un premier temps, la valeur entière 1 doit être convertie en la valeur réelle correspondante. Dans un second temps, cette valeur est inversée. Enfin, dans un troisième temps, la soustraction est effectuée :

```
LD    HL,1
LD    DE,Z2
CALL  ENTFLO    ;1 en flottant
LD    DH,Z2
CALL  NEGFLO    ; – 1 en flottant
LD    HL,Z1
LD    DE,Z2
CALL  ADDFLO    ;X^2 – 1
```

La racine, puis l'arc tangente de cette quantité sont calculées à l'aide des fonctions RAC et ATAN :

```
LD    HL,Z1
CALL  RACFLO    ;RAC(X^2 – 1)
LD    HL,Z1
CALL  ATAN      ;ATAN(RAC(X^2 – 1))
```

La fonction SIG renvoie le signe de l'argument :

```
LD    HL,Z3
CALL  SIG       ;Signe de X
```

La valeur renvoyée est :

- 0 si X est nul,
- 1 si X est positif,
- 255 si X est négatif.

Si X est négatif, la valeur 255 est convertie en – 1. Si X est positif ou nul, la valeur renvoyée n'est pas modifiée.

La quantité (SGN(X) – 1)*90 est calculée à l'aide des fonctions ADDFLO, ENTFLO et MULFLO selon le procédé habituel.

Enfin, la valeur finale est calculée en additionnant ATAN(RAC(X^2 – 1)) et (SGN(X) – 1)*90). Cette valeur est stockée dans la variable passée en entrée à l'aide du sous-programme FLODEHL :

```
FINPRE :  EQU   $
          LD    HL,Z2
```

```
LD    DE,(SAVHL)
CALL  FLODEHL      ;Résultat
```

Les sous-programmes **ZONE1** et **ZONE3** transfèrent respectivement les 5 octets pointés par **HL** dans Z1 et dans Z3.

Le sous-programme **FLODEHL** transfère les 5 octets pointés par **HL** à partir de l'adresse **(DE)**.

Reportez-vous aux programmes précédents pour avoir plus d'informations sur ces trois sous-programmes.

Si vous préférez utiliser un chargeur Basic, voici le listing et les données de checksum correspondantes :

```
1000 '--------------------------------
1010 ' Chargeur de la RSX ARC SECANTE
1020 '--------------------------------
1030 ' VERSION CPC 464
1040 '--------------------------------
1050 '
1060 FOR I=&9000 TO &90ED
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '--------------------------------
1130 ' Codes op de la RSX ARC SECANTE
1140 '--------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,1F,90,C3,2E,90,41
1180 DATA 53,45,C3,0,1,1A,90,21,16,90,CD,D1,BC,C9,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,D6,90,2A,0,90,CD,DF,90
1200 DATA 3E,FF,32,F7,B8,21,2,90,11,C,90,CD,61,BD,AF,21
1210 DATA 1,0,11,7,90,CD,40,BD,21,7,90,CD,6D,BD,21,2
1220 DATA 90,11,7,90,CD,58,BD,21,2,90,CD,79,BD,21,2,90
1230 DATA CD,91,BD,21,C,90,CD,70,BD,FE,FF,28,C,26,0,6F
1240 DATA 11,7,90,AF,CD,40,BD,18,10,AF,21,1,0,11,7,90
1250 DATA CD,40,BD,21,7,90,CD,6D,BD,AF,21,1,0,11,11,90
1260 DATA CD,40,BD,21,11,90,CD,6D,BD,21,7,90,11,11,90,CD
1270 DATA 58,BD,AF,21,5A,0,11,11,90,CD,40,BD,21,7,90,11
1280 DATA 11,90,CD,61,BD,11,2,90,CD,58,BD,21,7,90,ED,5B
1290 DATA 0,90,CD,E8,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1300 DATA C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0
```

```
0 73 3A 2E 40 4A 89 9F C6 2 C0 89 18 C4 79
```

```
1000 '--------------------------------
1010 ' Chargeur de la RSX ARC SECANTE
1020 '--------------------------------
1030 ' VERSION CPC 664
1040 '--------------------------------
1050 '
1060 FOR I=&9000 TO &90ED
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '--------------------------------
1130 ' Codes op de la RSX ARC SECANTE
1140 '--------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 15,90,C3,23,90,53,45,C3,0,1,1F,90,C3,2E,90,41
1180 DATA 53,45,C3,0,1,1A,90,21,16,90,CD,D1,BC,C9,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,D6,90,2A,0,90,CD,DF,90
1200 DATA 3E,FF,32,13,B1,21,2,90,11,C,90,CD,82,BD,AF,21
1210 DATA 1,0,11,7,90,CD,61,BD,21,7,90,CD,8E,BD,21,2
1220 DATA 90,11,7,90,CD,79,BD,21,2,90,CD,9A,BD,21,2,90
1230 DATA CD,B2,BD,21,C,90,CD,91,BD,FE,FF,28,C,26,0,6F
1240 DATA 11,7,90,AF,CD,61,BD,18,10,AF,21,1,0,11,7,90
1250 DATA CD,61,BD,21,7,90,CD,8E,BD,AF,21,1,0,11,11,90
1260 DATA CD,61,BD,21,11,90,CD,8E,BD,21,7,90,11,11,90,CD
1270 DATA 79,BD,AF,21,5A,0,11,11,90,CD,61,BD,21,7,90,11
1280 DATA 11,90,CD,82,BD,11,2,90,CD,79,BD,21,7,90,ED,5B
1290 DATA 0,90,CD,E8,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1300 DATA C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0
```

```
0 ED 3A 2E 75 8C CB E1 E7 44 3
CB 5A C4 79
```

```
1000 '--------------------------------
1010 ' Chargeur de la RSX ARC SECANTE
1020 '--------------------------------
1030 ' VERSION CPC 6128
1040 '--------------------------------
1050 '
1060 FOR I=&9000 TO &90ED
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '--------------------------------
1130 ' Codes op de la RSX ARC SECANTE
1140 '--------------------------------
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,0,0,0,0,0,1F,90,C3,2E,90,41
1180 DATA 53,45,C3,0,1,1A,90,21,16,90,CD,D1,BC,C9,DD,66
1190 DATA 1,DD,6E,0,22,0,90,CD,D6,90,2A,0,90,CD,DF,90
1200 DATA 3E,FF,32,13,B1,21,2,90,11,C,90,CD,85,BD,AF,21
1210 DATA 1,0,11,7,90,CD,64,BD,21,7,90,CD,91,BD,21,2
1220 DATA 90,11,7,90,CD,7C,BD,21,2,90,CD,9D,BD,21,2,90
1230 DATA CD,B5,BD,21,C,90,CD,94,BD,FE,FF,28,C,26,0,6F
1240 DATA 11,7,90,AF,CD,64,BD,18,10,AF,21,1,0,11,7,90
1250 DATA CD,64,BD,21,7,90,CD,91,BD,AF,21,1,0,11,11,90
1260 DATA CD,64,BD,21,11,90,CD,91,BD,21,7,90,11,11,90,CD
1270 DATA 7C,BD,AF,21,5A,0,11,11,90,CD,64,BD,21,7,90,11
1280 DATA 11,90,CD,85,BD,11,2,90,CD,7C,BD,21,7,90,ED,5B
1290 DATA 0,90,CD,E8,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1300 DATA C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0
```

```
0 73 3A 2E 78 92 D1 E7 EA 4A 9 D1 60 C4 79
```

Les RSX ¦COSEC (cosécante), ¦ACOSEC (arc cosécante), ¦COT (cotangente) et ¦ACOT (arc cotangente) utilisent les mêmes concepts que les RSX précédemment décrites.

Si, comme nous le pensons, vous avez compris le fonctionnement des RSX précédentes, les renseignements élémentaires qui suivent seront amplement suffisants à la parfaite compréhension du fonctionnement de ¦COSEC, ¦ACOSEC, ¦COT et ¦ACOT.