

9/4.9

Programmes divers

9/4.9.1

Coefficients du triangle de Pascal

Le triangle de Pascal donne les coefficients du polynôme d'ordre n :
 $(a + b)^n$

Dans ce chapitre, nous allons étudier la RSX: **PAS**, et le chargeur associé. Tous deux calculent les coefficients du polynôme pour un ordre compris entre 1 et 10.

Exemple :

Les coefficients du polynôme d'ordre 4 sont :

1 4 6 4 1

ce qui signifie que le polynôme

$(a + b)^4$

s'écrit après développement :

$a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$

COMMENT UTILISER LE PROGRAMME

Si vous désirez utiliser la version assembleur du programme, saisissez le listing suivant :

```

1          ORG  9000H
2          LOAD 9000H
3          ;-----
4          ; RSX PAS
5          ; Format : IPAS,ordre
6          ; Entree : ordre=puissance polynome
7          ; Sortie : Affichage des valeurs
8          ;-----
9          ;
10         ;
11         ;-----
12         ; Declaration des constantes
13         ; et des variables du programme
14         ;-----
15         ;
16         LOGEXT:  EQU  0BCD1H          ;KL LOG EXT
17         WRCHAR:  EQU  0BB5AH          ;TXT OUTPUT
18         ;
19         SAVA:    DS   1                ;Sauvegarde parametre
20         BUF:     DS   4                ;ZONE RAM POUR LOG EXT
21 9005 0A90      PTRTAB:  DW  TABLE      ;Pointeur TABLE
22 9007 C35990    JP   FASCAL            ;Calcul de la racine
23 900A 5041      TABLE:  DB  "PA"
24 900C D3        DB   "S"+80H

```

```

25 900D 00                                DB  0                                ;Fin de table
26                                         ;
27 900E 01010102 DATA:                   DB  1,1,1,2,1,1,3,3,1,
27 9012 01010303
27 9016 01010406
28 901A 04010105                           DB  4,1,1,5,10,10,5,1,
28 901E 0A0A0501
28 9022 01060F14
29 9026 0F060101                           DB  15,6,1,1,7,21,35,3
29 902A 07152323
29 902E 150701
30 9031 01081C38                           DB  1,8,28,56,70,56,28
30 9035 463B1C08
30 9039 01010924
31 903D 547E7E54                           DB  84,126,126,84,36,9
31 9041 24090101
31 9045 0A2D
32 9047 78D2FCD2                           DB  120,210,252,210,12
32 904B 782D0A01
33                                         ;
34                                         ;-----
35                                         ; Definition de la RSX
36                                         ;-----
37                                         ;
38 DEFRSX: EQU $                            ;Point d'entree

```

```
39 904F 010590      LD   BC, PTRTAB      ;Ptr table definition
40 9052 210190      LD   HL, BUF         ;Buffer pour LOG EXT
41 9055 CDD1BC      CALL LOGEXT         ;Definition de la RSX
42 9058 C9         RET
43                ;
44                ;
45                ;-----
46                ; Traitement de PASCAL
47                ;-----
48                ;
49                PASCAL: EQU $                ;Point d'entree
50                ;
51                ;-----
52                ; Lecture de la donnee passee
53                ;-----
54                ;
55 9059 DD7E00      LD   A, (IX+0)
56 905C 320090      LD   (SAVA), A      ;Sauvegarde
57                ;
58                ;-----
59                ; Affichage des valeurs
60                ;-----
61                ;
62 905F 47         LD   B, A
```

```
63 9060 AF          XOR  A
64                BOU1: EQU  $
65 9061 80          ADD  A,B
66 9062 10FD        DJNZ BOU1
67 9064 3D          DEC  A
68 9065 47          LD   B,A
69                ;
70                ; a ce niveau, B contient la
71                ; position du 1er element
72                ;
73 9066 3A0090      LD   A,(SAVA)
74 9069 4F          LD   C,A
75 906A 0C          INC  C                ;Nbre val a afficher
76                ;
77 906B 210E90      LD   HL,DATA
78 906E 1600        LD   D,0
79 9070 58          LD   E,B
80 9071 19          ADD  HL,DE
81 9072 41          LD   B,C
82                BOU2: EQU  $
83 9073 7E          LD   A,(HL)
84 9074 CD7B90      CALL AFFICHE                ;Aff 1 nombre
85 9077 23          INC  HL                ;prochaine memoire
86 9078 10F9        DJNZ BOU2
```

```
87 907A C9          RET
88                ;
89                ;-----
90                ; Zone des sous-programmes
91                ;-----
92                ;
93                ;-----
94                ; Affichage d'un facteur
95                ; du polynome
96                ;-----
97                ;
98                AFFICHE:   EQU   $
99 907B C5          PUSH BC
100 907C FEC8      CP      200
101 907E 3807      JR      C,I200
102 9080 D6C8      SUB     200
103 9082 0602      LD      B,2
104 9084 CD0091    CALL   AFCAR
105 9087 FE64      I200:   CP      100
106 9089 3807      JR      C,I100
107 908B D664      SUB     100
108 908D 0601      LD      B,1
109 908F CD0091    CALL   AFCAR
110 9092 FE5A      I100:   CP      90
111 9094 3807      JR      C,I90
```

112	9096	D65A		SUB	90
113	9098	0609		LD	B,9
114	909A	CD0091		CALL	AFCAR
115	909D	FE50	I90:	CP	80
116	909F	3807		JR	C,I80
117	90A1	D650		SUB	80
118	90A3	0608		LD	B,8
119	90A5	CD0091		CALL	AFCAR
120	90A8	FE46	I80:	CP	70
121	90AA	3807		JR	C,I70
122	90AC	D646		SUB	70
123	90AE	0607		LD	B,7
124	90B0	CD0091		CALL	AFCAR
125	90B3	FE3C	I70:	CP	60
126	90B5	3807		JR	C,I60
127	90B7	D63C		SUB	60
128	90B9	0606		LD	B,6
129	90BB	CD0091		CALL	AFCAR
130	90BE	FE32	I60:	CP	50
131	90C0	3807		JR	C,I50
132	90C2	D632		SUB	50
133	90C4	0605		LD	B,5
134	90C6	CD0091		CALL	AFCAR
135	90C9	FE28	I50:	CP	40

```
136 90CB 3807          JR   C,I40
137 90CD D628          SUB  40
138 90CF 0604          LD   B,4
139 90D1 CD0091        CALL AFCAR
140 90D4 FE1E          I40:  CP   30
141 90D6 3807          JR   C,I30
142 90D8 D61E          SUB  30
143 90DA 0603          LD   B,3
144 90DC CD0091        CALL AFCAR
145 90DF FE14          I30:  CP   20
146 90E1 3807          JR   C,I20
147 90E3 D614          SUB  20
148 90E5 0602          LD   B,2
149 90E7 CD0091        CALL AFCAR
150 90EA FE0A          I20:  CP   10
151 90EC 3807          JR   C,I10
152 90EE D60A          SUB  10
153 90F0 0601          LD   B,1
154 90F2 CD0091        CALL AFCAR
155 90F5 47            I10:  LD   B,A
156 90F6 CD0091        CALL AFCAR
157 90F9 3E20          LD   A,32
158 90FB CD5ABB        CALL WRCHAR          ;Separateur
159 90FE C1            POP  BC
```



```

160 90FF C9          RET
161
162                AFCAR:    EQU  $
163 9100 F5          PUSH AF
164 9101 78          LD  A,B
165 9102 C630        ADD  A,48
166 9104 CD5ABB      CALL WRCHAR          ;Affichage 1 caract
167 9107 F1          POP  AF
168 9108 C9          RET
169
170                END

```

Installez la RSX en tapant sous Basic

CALL &904F

Pour obtenir les coefficients du polynôme, il vous suffit de taper
!PAS,n

où n est compris entre 1 et 10.

Par exemple, l'instruction

!PAS,6

affichera

1 6 15 20 15 6 1

qui sont les coefficients du polynôme d'ordre 6.

Si vous désirez utiliser la version Basic du programme, saisissez le listing suivant :

```

1000 REM -----
1010 REM Chargeur BASIC de la RSX !PAS
1020 REM -----
1030 REM
1040 FOR i=&9000 TO &9108
1050   READ a$
1060   a$="&"+a$
1070   POKE i,VAL(a$)
1080 NEXT i
1090 CALL &904F
1100 END
1110 REM - - - - -
1120 REM Donnees de la RSX
1130 REM - - - - -
1140 REM
1150 DATA 21,69,90,22,65,A,90,C3,59,90,50,41,D3,0,1,1
1160 DATA 1,2,1,1,3,3,1,1,4,6,4,1,1,5,A,A
1170 DATA 5,1,1,6,F,14,F,6,1,1,7,15,23,23,15,7
1180 DATA 1,1,8,1C,38,46,38,1C,8,1,1,9,24,54,7E,7E
1190 DATA 54,24,9,1,1,A,2D,7B,D2,FC,D2,78,2D,A,1,1
1200 DATA 5,90,21,1,90,CD,D1,BC,C9,DD,7E,0,32,0,90,47
1210 DATA AF,80,10,FD,3D,47,3A,0,90,4F,C,21,E,90,16,0
1220 DATA 58,19,41,7E,CD,7B,90,23,10,F9,C9,C5,FE,C8,38,7
1230 DATA D6,C8,6,2,CD,0,91,FE,64,38,7,D6,64,6,1,CD
1240 DATA 0,91,FE,5A,38,7,D6,5A,6,9,CD,0,91,FE,50,38
1250 DATA 7,D6,50,6,8,CD,0,91,FE,46,38,7,D6,46,6,7
1260 DATA CD,0,91,FE,3C,38,7,D6,3C,6,6,CD,0,91,FE,32
1270 DATA 38,7,D6,32,6,5,CD,0,91,FE,28,38,7,D6,28,6
1280 DATA 4,CD,0,91,FE,1E,38,7,D6,1E,6,3,CD,0,91,FE
1290 DATA 14,38,7,D6,14,6,2,CD,0,91,FE,A,38,7,D6,A
1300 DATA 6,1,CD,0,91,47,CD,0,91,38,20,CD,5A,BB,C1,C9
1310 DATA F5,78,C6,30,CD,5A,BB,F1,C9,0,0,0,0,0,0,0

```

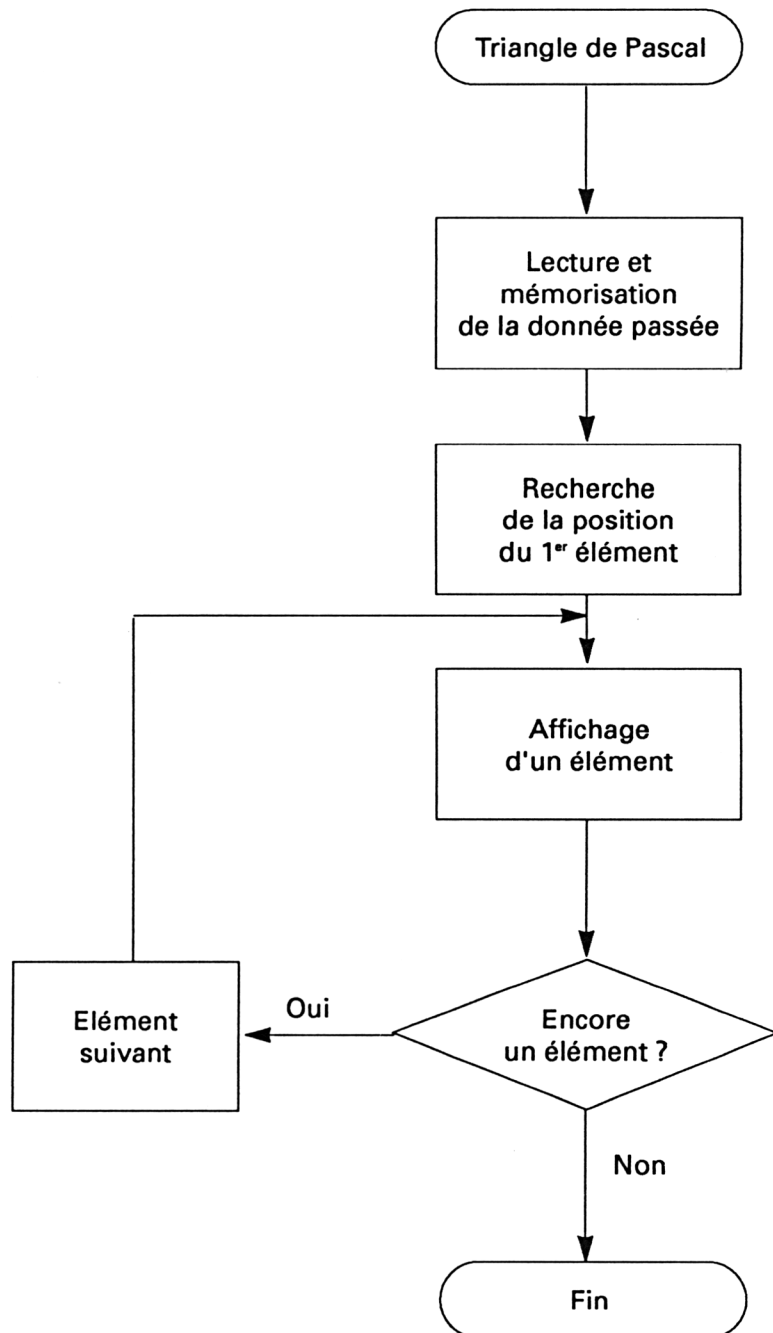
et utilisez le programme de checksum pour vérifier que les lignes de DATA entrées sont conformes au listing Weka :

```
52 36 C5 81 87 D4 BE CE B9 51 4A B9 1E 1C CE DA 5
```

Lancez le programme. La RSX est automatiquement installée. Il vous suffit de l'activer par une instruction !PAS pour connaître les coefficients du polynôme (voir ci-dessus).

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Les premières lignes du programme sont consacrées aux constantes et variables.

Vous y reconnaissez les macros :

- LOGEXT (ligne 16) qui permet de définir une nouvelle RSX ;
- WRCHAR qui permet d'afficher un caractère sur l'écran ;
- les zones mémoires habituelles aux RSX : BUF, PTRTAB et TABLE.

L'octet SAVA est utilisé pour sauvegarder le nombre passé à la RSX.

Ces constantes et variables sont suivies des coefficients du polynôme (lignes 27 à 32). En effet, le programme ne calcule pas ces coefficients. Il se contente de les lire en mémoire.

Le programme se poursuit par la routine d'installation de la RSX lignes 38 à 42. Cette routine est très classique. Nous n'y reviendrons pas.

Lorsque le mot « !PAS » est reconnu par le Basic, il donne le contrôle au programme situé à l'étiquette PASCAL (ligne 49).

Etudions le fonctionnement de ce programme.

Le paramètre entier passé par valeur à la RSX est stocké dans la mémoire SAVA :

```
LD A, (IX+0)
LD (SAVA), A
```

Les lignes suivantes calculent (en fonction du paramètre) la position de la première donnée à afficher.

Ce calcul se fait à partir de la constatation suivante :

- le polynôme de degré 1 est composé de deux monômes ;
- le polynôme de degré 2 est composé de 3 monômes ;
- ...
- le polynôme de degré N est composé de N+1 monômes.

Les coefficients étant stockés à des positions mémoires consécutives, le premier coefficient du polynôme de degré N se trouvera à la position

$$N-1 + N-2 + \dots + 2$$

Les instructions qui effectuent ce calcul sont les suivantes :

```
LD B, A      --> Degré du polynôme
XOR A
BOU1 EQU $
ADD A, B     --> Une des additions
DJNZ BOU1   --> Boucle si ce n'est pas la dernière ?
DEC A       --> Le résultat est décrémenté de 1
```

La position du premier coefficient étant connue, il suffit de lui ajouter l'adresse du premier coefficient du degré 1 pour avoir son adresse en mémoire.

```
LD    B, A
...
LD    HL, DATA
LD    D, 0
LD    E, B
ADD   HL, DE          --> Adresse du coefficient à afficher
```

Les instructions qui suivent appellent répétitivement le sous-programme **AFFICHE** qui affiche un des coefficients sur l'écran :

```
BOU2: EQU $
      LD    A, (HL)    --> Lecture d'un coefficient
      CALL AFFICHE    --> Affichage
      INC  HL          --> Elément suivant
      DJNZ BOU2       --> Boucle si ce n'est pas le dernier
```

Le programme se poursuit par la zone des sous-programmes.

Le sous-programme **AFFICHE** fait appel à la routine **AFCAR** pour afficher le ou les chiffres qui constituent le coefficient.

Etudions son fonctionnement.

Le degré maximal (10) n'a pas été choisi au hasard. Il correspond au dernier degré pour lequel le coefficient maximum est inférieur à 255, et donc, codable sur un octet.

Le coefficient à afficher est successivement comparé à 200, 100, 90, ..., 20 et 10.

S'il est supérieur à l'une de ces valeurs, elle lui est soustraite et donne lieu à l'affichage d'un chiffre. Pour clarifier ces propos, considérons l'exemple suivant : le coefficient à afficher est égal à 212.

Ce nombre est comparé à 200. Il est supérieur. Donc, on lui soustrait 200, et on affiche un 2. La valeur résiduelle (12) est ensuite comparée à 100, 90, etc., jusqu'à 10. Elle est supérieure à 10. Donc, on lui soustrait 10 et on affiche 1. La valeur résiduelle (2) est ensuite affichée.

Finalement, les chiffres 2, 1 et 2 ont été affichés, ce qui correspond bien à l'effet souhaité.

Les lignes 100 à 158 réalisent cette batterie de tests.

Remarquez les instructions **PUSH BC** et **POP BC** lignes 99 et 159 qui laissent la paire de registres **BC** intacte après une exécution du sous-programme **AFFICHE**. Ces instructions sont fondamentales. Si elles

n'existaient pas, le registre B qui indique le nombre de chiffres à afficher serait détruit. Le dernier sous-programme a pour nom **AFCAR**. Il convertit un chiffre en un code ASCII en lui ajoutant 48, et l'affiche sur l'écran à l'aide de la macro **WRCHAR** :

LD	A, B	--> Le chiffre à afficher est stocké dans A
ADD	A, 48	--> Il est converti en un code ASCII
CALL	WRCHAR	--> Et affiché sur l'écran

Remarquez là aussi les instructions **PUSH AF** et **POP AF** lignes 163 et 167 qui évitent la destruction des registres B et C.