

TP Assembleur - E2I3

Léo Dubus - tp3-1.asm

```

1  CODE    $1000
2  DATA    $2000
3  PUTCHAR $EE86
4
5          ORG    DATA
6  CHAINE   DC.B    "The Game",10,0
7
8          ORG    CODE
9  LDY      #CHAINE    ;On récupère l'adresse de la chaine
10                     ;à afficher dans Y
11  BSR      AFFICH    ;On effectue l'affichage
12  SWI                      ;On arrête
13
14
15  AFFICH   LDAB      1,Y+    ;On charge le premier caractère
16                     ;dans le registre B
17
18          PSHY          ;On backup la valeur de Y dans la pile
19                     ;pour éviter un écrasement
20
21          JSR      TEMPO    ;On fait une tempo
22          JSR      [PUTCHAR,PCR] ;On affiche le premier caractère
23                     ;situé dans B
24
25          PULY          ;On récupère l'adresse du prochain caractère dans Y
26          BNE      AFFICH    ;Si on n'obtient pas le caractère nul, on continue.
27          RTS          ;fin de sous routine
28
29  ;*-- TEMPORISATION -----
30  TEMPO    LDAA      #$FA    ;On commence la tempo
31  REC2     LDX      #$0FFF
32  REC1     DEX
33          BNE      REC1
34          DECA
35          BNE      REC2
36          NOP          ;Tempo fini
37          RTS
38  ;*-----

```

```

1  CODE    $1000
2  DATA   $2000
3  GETCHAR $EE84
4
5          ORG     DATA
6  VAR     DS.B   1
7
8          ORG     CODE
9  LDX     #VAR    ;On récupère l'adresse de la variable dans X
10 BSR     SCRUT   ;On commence la scrutation des caractères entrés
11                    ;par l'utilisateur
12
13          SWI
14
15  SCRUT   CLRB    ;On nettoie B pour être sûr du caractère qui va être
16                    ;récupéré
17          JSR     [GETCHAR,PCR] ;On récupère le caractère
18  CMPB    #'+'    ;On le compare avec "+"
19  BEQ     INCREM   ;Si oui, incrémentation de la variable
20  CMPB    #'-'    ;Comparaison avec "-"
21  BEQ     DECREM   ;Si oui, on décrémente
22  CMPB    #'$1B   ;Comparaison avec la touche ESC
23  BNE     SCRUT    ;Si non, on recommence
24          RTS     ;sinon on arrête
25
26  INCREM  INC     0,X ;On incrémente VAR
27          BRA     SCRUT ;On recommence à scruter
28
29  DECREM  DEC     0,X ;On décrémente VAR
30          BRA     SCRUT ;On recommence à scruter
31

```

```

1  CODE    $1000
2  DATA   $2000
3  GETCHAR $EE84
4  PUTCHAR $EE86
5  SCIOSR1 $CC
6  SCIODRL $CF
7
8          ORG     DATA
9  VAR     DS.B    1
10
11         ORG     CODE
12         LDAA    #$02      ;On charge la valeur 2 dans A pour la stocker
13                               ;dans la mémoire
14         STAA    VAR      ;Il s'agit de la valeur de départ
15         BSR     SCRUT    ;On effectue la structation avec la fonction AUVOL
16         SWI
17
18  SCRUT   BSR     AUVOL
19         CMPB    #'+'      ;Comme précédemment, on compare le caractère entré
20         BEQ     INCREM    ;Et on incrémente la variable
21         CMPB    #'-'
22         BEQ     DECREM
23         CMPB    #$1B
24         BEQ     FIN
25
26         LDAB    #'I'      ;Caractère qui va clignoter
27         JSR     [PUTCHAR,PCR] ;On fait apparaitre le caractère
28         BSR     TEMPO      ;On attend
29         LDAB    #$08      ;On charge le caractère BACKSPACE
30         JSR     [PUTCHAR,PCR] ;On le fait apparaitre
31         BSR     TEMPO      ;On attend
32
33         BRA     SCRUT      ;Et on recommence indéfiniment
34         RTS
35
36  INCREM  INC      VAR
37         BRA     SCRUT
38
39  DECREM  DEC      VAR
40         BRA     SCRUT
41
42
43  ;*-- TEMPORISATION -----
44  TEMPO   LDAA    VAR      ;On commence la tempo
45  REC2    LDY     #$FFFF    ;Tempo à 50 000
46  REC1    DEY
47         BNE     REC1
48         DECA
49         BNE     REC2
50         NOP      ;Tempo fini
51         RTS

```

```
52 ;*-----
53
54 AUVOL    BRCLR    SCIOSR1,#$20,VIDE
55          LDAB     SCIODRL ;
56          BRA      FIN
57
58 VIDE     CLR      CLRB
59
60 FIN      RTS
```

```

1  CODE    $1000
2  DATA   $2000
3  VALEUR  $3000
4  PRINTF  $EE88
5  OUT2HEX
6
7          ORG     DATA
8  VAR     DC.B    'Valeur : %d',10,0
9
10         ORG     CODE
11  CLRA                    ;On clear A parce qu'on veut uniquement la valeur de
12                          ;£3000 et printf prend des mots en argument
13  LDAB     VALEUR        ;On charge B avec la valeur voulue en £3000
14  PSHD                    ;On passe la valeur totale en paramètre par la pile
15  LDD      #VAR          ;On charge la chaine de caractère pour le printf
16  JSR      [PRINTF,PCR]   ;On effectue le printf
17  PULD                    ;On récupère notre valeur.
18
19
20
21  SWI

```

```

1  CODE    $1000
2  DATA   $2000
3  VALEUR  $3000
4  PRINTF  $EE88
5  GETCHAR $EE84
6  PUTCHAR $EE86
7
8          ORG     DATA
9  AFF      DC.B    'Valeur : %d',10,0
10 TABL     DC.W    PHRASE1,PHRASE2,PHRASE3,DEFAULT
11 PHRASE1   DC.B    'Salut',10,0
12 PHRASE2   DC.B    'cest',10,0
13 PHRASE3   DC.B    'cool',10,0
14 DEFAULT  DC.B    'NON',10,0
15
16
17          ORG     CODE
18          LDX     #TABL
19          CLRA
20          PSHX
21          JSR     [GETCHAR,PCR]
22          SUBB    #2F
23          PULX
24          ASLB
25          ABX
26
27          LDAB    0,X
28
29          PSHX
30 AFFICH    JSR     [PUTCHAR,PCR]
31          PULX
32          BNE     AFFICH
33
34          SWI

```