



Architecture Élémentaire: Fiche de TP 2

Structures de Contrôle

octobre 2001

1 Introduction

Considérons un programme Assembleur dont le squelette est donné figure 1. Voici en exemple, comment sont implémentés différentes structures de contrôle.

1. Si Alors

```
; =====
; si X>Y alors
; <instructions>
; fsi
    mov AX, X
    mov BX, Y
    cmp AX, BX
    jna fsi1 ; not above
; <instructions>
fsi1:
; =====
; si X<Y alors
; <instructions>
; fsi
    mov AX, X
    mov BX, Y
    cmp AX, BX
    jnb fsi2 ; not below
; <instructions>
fsi2:
; =====
; si X=Y alors
; <instructions>
; fsi
    mov AX, X
    mov BX, Y
    cmp AX, BX
    jne fsi3
; <instructions>
fsi3:
; =====
; si X=0 alors
; <instructions>
; fsi
```

```

        mov AX, X
        and AX, AX    ; plus rapide que cmp AX, 0
        jnz fsi4
    ; <instructions>
    fsi4:

```

```

; Programme skeleton.asm structure de controle
; -----
; Directives
; -----
    DOSSEG
    .MODEL small
    .STACK 200h
; -----
; Données
; -----
    .DATA
    car db ?    ; car est un octet (define byte)
    X dw ?      ; X est un mot (define word)
    Y dw ?      ; Y est un mot
; -----
; Code
; -----
    .CODE
Begin:    ; point d'entrée du programme
    mov AX,@data
    mov DS,AX    ; on initialise le segment de données
; =====
; partie à compléter
; =====
    mov AH, 4Ch
    int 21h      ; sortie du programme
    END Begin

```

Figure 1: Squelette du programme Assembleur

2. Si Alors Sinon

```

; =====
; si X=0 alors
;   <instructions_alors>
; sinon
;   <instruction_sinon>
; fsi
    mov AX, X
    and AX, AX
    jne sinon
; <instructions_alors>
    jmp fsi5
sinon:
; <instructions_sinon>
    fsi5:

```

3. Pour

```

; =====
; pour CX décroissant de X à 1 faire
; <instructions>
; fait
    mov CX, X
    and CX, CX
    je fait1
boucle1:
; <instructions>
    loop boucle1;
fait1:
; =====
; pour AX croissant de Y à X faire
; <instructions>
; fait
    mov BX, X
    mov AX, Y
boucle2:
    cmp AX, BX
    ja fait2
; <instructions>
    inc AX
    JMP boucle2
fait2:

```

4. Tant que

```

; =====
; tantque <test>
; <instructions>
; fait
boucle3:
; <test>
    jxx fait3 ; jump conditionnel adéquat si le test n'est
                ; pas satisfait
; <instructions>
    jmp boucle3
fait3:

```

5. Répéter

```

; =====
; répéter
; <instructions>
; tantque <test>
boucle4:
; <instructions>
; <test>
    jxx boucle4 ; jumps conditionnel adéquat si le test est satisfait

```

2 Travail à faire

Question 2.1 : *Écrire un programme qui lit au clavier un entier N codé sur un caractère (de 0 à 9 donc) et affiche une ligne de N caractères '*'.*

Question 2.2 : *Écrire un programme qui lit au clavier un entier N codé sur plusieurs caractères (saisie finie par "retour chariot" — code ASCII 13) et affiche une ligne de N caractères '*'.*

Question 2.3 : *Écrire un programme qui lit au clavier un entier N codé sur plusieurs caractères (saisie finie par "retour chariot") et affiche un carré composé de '*' de côté N .*

Question 2.4 : *Écrire un programme qui lit au clavier un entier N codé sur plusieurs caractères (saisie finie par "retour chariot") ainsi qu'un caractère désignant un type de figure et qui dessine la figure correspondante comme illustré figure 2 (pour chaque type, l'exemple pour $N = 4$ et $N = 5$ est donné).*

type 'c'	type 't'	type 'i'
*****	*	*
**** *****	* **	* ***
**** *****	** ***	** ****
**** *****	*** ****	*** *****
**** *****	**** *****	**** *****

type 's'

**** **
** *
** ***
**** *****

Figure 2: Types de figures avec exemple $N = 4$ et $N = 5$.