

Explication : Chap 3 Assembleur / Code machine

1) La syntaxe de déclaration des segments est de la forme suivante :

<Nom_Segment> *SEGMENT*

Contenu du segment

<Nom_Segment> *ENDS*

<Nom_Segment> : est un nom au choix du programmeur.

Exemple :

MesVar *SEGMENT*

 ; Commentaire

 ; Mes instructions

MesVar *ENDS*

MonCode *SEGMENT*

ASSUME DS:MesVar, CS:MonCode

Debut : MOV AX, @MesVar ; *AX ← l'adresse logique de base du segment MesVar*

 MOV DS, AX

 MOV AX, @MonCode ; *AX ← l'adresse logique de base du segment MonCode*

 MOV CS, AX

 ; *Les instruction de mon programme : La somme de AX et BX*

 ADD AX, BX ; *AX ← AX + BX*

MonCode *ENDS*

END Debut

2) La syntaxe de déclaration des variables :

2.1) Variables simples :

<NomVariable> TYPE ValeurInitialisation1,, ValeurInitialisationN

Exemple :

X	DB	12H, 'A', 2+5 ; X de type BYTE (8BITS)	7	...	0
Y	DW	1234H, ? ; Y de type WORD (16BITS)			
Z	DD	56H, ? ; Z de type DOUBLE WORD (32BITS)			
			X		
			X+1		
			X+2		
			Y		
			Y+1		
			Z		
			Z+1		

—
—
...
12H
'A'
2+5=7
34H
12H
?
?
56H
00H
00H
00H
?
?
?
?

3) Instructions assembleurs :

Elles sont de la forme : Operateur ArgDest (Argument destinataire), ArgSrce (Argument source)

3.1) Instructions de déplacement des données :

a) MOV ArgDest, ArgSrce ; ArgDest \leftarrow ArgSrce

MOV AL, 20H ; AL = 20H

PUSH ArgSrce ; Empiler ArgSrce = Placer La valeur de ArgSrce dans la pile

POP ArgDest ; Depiler la valeur du sommet de Pile et l'affecter à ArgDest : ArgDest \leftarrow ; Sommet de pile

Exemple d'écriture d'un programme calculant la somme de deux variables X et Y

```
MyData      SEGMENT

    X        DB    54H

    Y        DB    78H

MyData      ENDS

MyCode      SEGMENT

    ASSUME    DS :MyData, CS : MyCode

Main : MOV    AX,    @MyData

        MOV    DS,    AX

        MOV    AX,    @MyCode

        MOV    CS,    AX

        ; LA SOMME

        MOV    AX,    Y        ; AX ← Y

        ADD    AX,    X        ; AX ← AX=Y + X

MyCode      ENDS

END    Main
```