

---

**Architecture des ordinateurs**  
– Assembleur –

---

**Exercice 1 :** Écrire un programme en assembleur 8086 qui introduit une chaîne de caractères via le clavier et l’affiche à l’envers.

---

**Solution**

---

```
org 100h
mov cl, 0 ; pour compter le nombre de caractères lu

Lecture:
mov ah, 08h ; permet de lire un caractère au clavier
int 21h
mov bl, al ; le caractère lu sera stocké en bl

cmp bl, 0Dh ; comparaison avec l'ASCII du Retour à la ligne
je afficher ; si ils sont égaux, il faut afficher
add cl, 1 ; compter le nombre de caractères lus
push bx ; sinon, mettre le contenu du registre dans la pile
; 16 bits
jmp Lecture ; relire un nouveau caractère

afficher:
mov ah, 02h ; chargement des instructions d'affichage
pop dx ; depiler la pile et mettre les 16 bits de la pile dans dx,
; le dx, contient le caractère à afficher.
int 21h
loop afficher ; boucler sur afficher cl fois.

int 20h ; fin du programme
```

**Exercice 2 :** Écrire un programme assembleur 8086/8088 qui calcule le  $n^{\text{ème}}$  terme de suite suivante (sans chercher à calculer sa forme explicite).

$$\begin{aligned}U_0 &= 1 \\U_n &= 3 \times U_{n-1} + n\end{aligned}$$

- Déclarer des variables entières pour stocker la valeur de  $n$ , la valeur de  $U_0$ , et le résultat.
- Déclarer une variable pour stocker la valeur constante 3.
- Ajouter à votre programme des instructions pour afficher le résultat.
- Quelle valeur sera affichée lorsque  $n$  vaut 3.

---

**Solution**

---

```

org 100h

mov cl,n
mov bl,0
mov al,U0
Suite:
    mul c    ; muplitation de al par 3
    inc bl   ; incrimente bl
    add al,bl ; ajoute bl a al
loop Suite
    ; pour n =3 le resultat est 2D en base 16
    ; c'est donc 45 en base 10

;;: l'affichage
mov dl, al
mov ah,02h
int 21h
    ; le caracte affiche sera "-" le code ascii de 45
int 20h

U0 db 1
n db 3 ; la valeur de n
c db 3 ; declaration de la constante 3
ret

```

### Exercice 3 : Langage machine (3 pts)

Ci-dessous un programme en langage machine, implanté à l'adresse 0100H:

```
B8 B8 00 A3 10 10 B8 05 00 2D 03 00
```

A l'aide de la table d'instructions (voir Table 1), donner la transcription langage symbolique de ce programme et expliquer en français ce qu'il fait. L'adresse de début de chaque instruction est indiquée à gauche (en hexadécimal).

Adresse	Contenu MP	Langage Symbolique	Explication en français
0100	B8 B8 00	...	...
...	...	...	...
...	...	...	...

**Solution**

```

-a100
19A8:0100 mov ax,B8
19A8:0103 mov [1010],ax
19A8:0106 mov ax,5
19A8:0109 add ax,[1010]
19A8:010D sub ax,3
19A8:0110
-u 100
19A8:0100 B8B800 MOV AX,00B8
19A8:0103 A31010 MOV [1010],AX
19A8:0106 B80500 MOV AX,0005
19A8:0109 03061010 ADD AX,[1010]
19A8:010D 2D0300 SUB AX,0003
19A8:0110 B80500 MOV AX,0005
19A8:0113 0000 ADD [BX+SI],AL
19A8:0115 0000 ADD [BX+SI],AL
19A8:0117 0000 ADD [BX+SI],AL
19A8:0119 0000 ADD [BX+SI],AL
19A8:011B 0034 ADD [SI],DH
19A8:011D 00971900 ADD [BX+0019],DL
-

```

Adresse	Contenu MP	Langage Symbolique	Explication en français
0100	B8 B8 00	MOV AX,B8	Charger le registre AX par la valeur B8
103	A31010	MOV [1010],AX	Enregistrer le contenu de AX dans la zone mémoire [1010]
106	B80500	MOV AX,5	Charger le registre AX par la valeur B8
109	03061010	ADD AX,[1010]	Additionner AX avec le contenu de la zone mémoire [1010]. Le résultat sera stocké en AX
10D	020300	SUB AX, 3	Soustraire de AX la valeur 3 et le résultat sera en AX.

#### Exercice 4 : Produit des $n$ premiers termes

Écrire un programme assembleur qui permet de calculer et d'afficher le produit des  $n$  premiers termes non nuls ( $\prod_{i=1}^n i$ ),  $n$  étant un nombre entier strictement positif saisi au clavier. Cela revient à écrire les parties de code assembleur suivantes:

- Ecrire le code pour lire la valeur de  $n$  au clavier.
- Tester la validité de cette valeur ( $n > 0$ ).
- Calculer le produit si  $n$  est valide sinon afficher un message d'erreur.

#### Solution

Il est à remarquer que  $\prod_{i=1}^n i = n!$

```

org 100h

; lecture du nombre
mov ah, 08h
int 21h
; le caractere lu est enregistre dans al

; calcul du produit
mov cl,al
mov al,1
Prod:
    mul cl
    dec cl
    loop prod

; affichage du resultat
mov ah, 02h
mov dl,al
int 21h

; fin du programme
int 20h

```

### Exercice 5 : Langage machine

Ci-dessous un programme en langage machine, implanté à l'adresse mémoire 0x0100:

```

101110000000000100000000 101110010000010100000000 1111011111100001
01001001 100000111111100100000001 0111011111111000 1000101011010000
1011010000000010 1100110100100001 1100110100100000

```

A l'aide de la table d'instructions (voir Table 1), donner la transcription en langage symbolique de ce programme et expliquer en français ce qu'il fait. L'adresse de début de chaque instruction est indiquée à gauche (en hexadécimal).

Adresse	Contenu MP	Langage Symbolique	Explication en français
0100	...	...	...
...	...	...	...
...	...	...	...

### Solution

```

1011 1000 0000 0001 0000 0000 1011 1001 0000 0101 0000 0000 1111 0111 1110 0001

```

$\overbrace{0100}^{B}$   $\overbrace{1001}^{8}$   $\overbrace{1000}^{F}$   $\overbrace{0011}^{7}$   $\overbrace{1111}^{E}$   $\overbrace{1001}^{4}$   $\overbrace{0000}^{9}$   $\overbrace{0001}^{3}$   $\overbrace{0111}^{0}$   $\overbrace{0111}^{1}$   $\overbrace{1111}^{0}$   $\overbrace{1000}^{5}$   $\overbrace{1000}^{0}$   $\overbrace{1010}^{B}$   $\overbrace{1101}^{4}$   $\overbrace{0000}^{0}$   
 $\overbrace{1011}^{1}$   $\overbrace{0100}^{0}$   $\overbrace{0000}^{0}$   $\overbrace{0010}^{1}$   $\overbrace{1100}^{8}$   $\overbrace{1101}^{3}$   $\overbrace{0010}^{F}$   $\overbrace{0001}^{8}$   $\overbrace{1100}^{B}$   $\overbrace{1101}^{4}$   $\overbrace{0010}^{5}$   $\overbrace{0000}^{0}$

Transcription en hexadécimal est :

B80100 B90500 F7E1 49 83F901 77F8 8AD0 B402 CD21 CD20

Adresse	Contenu MP	Langage Symbolique	Explication en français
0100	B80100	MOV AX, 0001h	charger le registre AX avec la valeur 1h
0103	B90500	MOV CX, 0005h	charger le registre CX avec la valeur 5h
0106	F7E1	MUL CX	Multiplier AX avec CX, le résultat est dans AX
0108	49	DEC CX	Décrémenter CX
0109	83F901	CMP CX, 01h	Comparer CX avec 01
010D	77F8	JNBE 0106h	Sauter à l'adresse 106h
010F	8AD0	MOV DL, AL	Afficher le résultat. Charger DL avec AL
0111	B402	MOV AH, 02h	Charger AH avec 02h
0113	CD21	INT 021h	Appel de BIOS pour afficher le résultat
0115	CD20	INT 020h	Fin du programme

Symbole	Code Op.	Octets	Opération
MOV AX, valeur	B8	3	$AX \leftarrow \text{valeur}$
MOV AX, [ adr ]	A1	3	$AX \leftarrow \text{contenu de l'adresse adr.}$
MOV [ adr ], AX	A3	3	range AX à l'adresse adr.
MOV CX, valeur	B9	3	$CX \leftarrow \text{valeur}$
MOV DL, AL	8AD0	2	$DL \leftarrow AL$
MOV AH, valeur	B4	2	$AH \leftarrow \text{valeur}$
ADD AX, valeur	05	3	$AX \leftarrow AX + \text{valeur}$
ADD AX, [ adr ]	03 06	4	$AX \leftarrow AX + \text{contenu de adr.}$
SUB AX, valeur	2D	3	$AX \leftarrow AX - \text{valeur}$
SUB AX, [ adr ]	2B 06	4	$AX \leftarrow AX - \text{contenu de adr.}$
MUL CX	F7E1	2	$AX \leftarrow AX \times CX$
SHR AX, 1	D1 E8	2	décale AX à droite.
SHL AX, 1	D1 E0	2	décale AX à gauche.
INC AX	40	1	$AX \leftarrow AX + 1$
DEC AX	48	1	$AX \leftarrow AX - 1$
DEC CX	49	1	$CX \leftarrow CX - 1$
CMP AX, valeur	3D	3	compare AX et valeur.
CMP AX, [ adr ]	3B 06	4	compare AX et contenu de adr.
CMP CX, valeur	83 F9	4	compare CX et valeur.
JMP adr	EB	2	saut inconditionnel (adr. relatif).
JE adr	74	2	saut si =
JNE adr	75	2	saut si $\neq$
JNBE adr	77	2	saut si $>$
JG adr	7F	2	saut si $>$
JLE adr	7E	2	saut si $\leq$
JA adr		2	saut si $CF = 0$
JB adr			saut si $CF = 1$
INT 20	CD 20	2	Fin du programme.
INT 21	CD 21	2	Appel du BIOS DI et retour au DOS

Tableau 1: Tableau des instructions