

### EXERCICE N°1:

Écrire et exécuter le programme assembleur 8086 pour chaque expressions arithmétiques ci-dessous :

- ✓  $Ax = Bx + Cx - Dx$
- ✓  $Ax = Bx * Cx + 300h$
- ✓  $Ax = (Bx * Cx) / Dx + SI$
- ✓  $[Bx] = (Ax * Cx) + (Bx * Dx)$

### EXERCICE N°2:

Donner un programme assembleur 8086 qui permet de déterminer le maximum entre les registres AX, BX et CX. La première étape consiste à lire 03 mots à partir de l'adresse 100h et les mettre dans les registres AX, BX et CX dans cet ordre. Sauvegarder le maximum dans l'adresse 200h.

### EXERCICE N°3:

Donner un programme assembleur 8086 qui permet de déterminer le minimum entre les registres AL, BL et CL. La première étape consiste à lire 03 cases à partir de l'adresse 100h et les mettre dans les registres AL, BL et CL dans cet ordre. Sauvegarder le minimum dans l'adresse 200h.

### EXERCICE N°4:

Donner un programme assembleur 8086 qui permet de faire le suivant:

```
if (x > 31) {  
    cx = ax/bx;  
}
```

tel que: x= [100h]

### EXERCICE N°5:

Donner un programme assembleur 8086 qui permet de faire le suivant:

```
if (x > 5) {  
    cl = ax/bl - x;  
}
```

```
} else {  
    cl = ax/bl + x;  
}
```

tel que: x= [100h], et bl = [102h]+1

### EXERCICE N°6:

Donner un programme assembleur 8086 qui permet de faire le suivant:

```
i= 0;
```

```
while (i < 10){  
    x = x + 3;    i ++;  
}
```

tel que: x= [100h]

### EXERCICE N°7:

Donner un programme assembleur 8086 qui permet de faire le suivant:

```
int a=0;
```

```
for (int i=0 ;i <10; i++)
```

```
    a=a+i*2;
```

stocker a dans [100h].

### EXERCICE N°8:

Donner un programme assembleur 8086 qui permet de faire le suivant:

```
i=0;
```

```
while (i < 10){
```

```
    if (x > 5) {
```

```
        c = a/b - x; } else {
```

```
        c = a/b + x;
```

```
    }
```

```
    i++;
```

```
}
```

tel que: x= [100h], (a,b,c)=([200h], [201h], [202h]).

### EXERCICE N°9:

Donner un programme assembleur 8086 qui permet de déterminer le maximum d'un tableau. La taille de chaque élément est un octet et la taille du tableau est stockée dans l'adresse 100h. Le tableau est stocké à partir de l'adresse 200h. Sauvegarder le résultat dans la case de l'adresse 300h.

### EXERCICE N°10:

Donner un programme assembleur 8086 qui permet de déterminer le nombre de **zéro** dans un tableau. La taille de chaque élément est un octet et le nombre des éléments est stocké dans l'adresse 100h. Le tableau est stocké à partir de l'adresse 200h. Sauvegarder le résultat dans la case de l'adresse 300h.

### EXERCICE N°11:

Donner un programme assembleur 8086 qui permet d'effectuer la somme S suivante. Le nombre n est stocké dans l'adresse 100h. Le résultat doit être stocker dans l'adresse 300h.

$$S = \sum_{i=1}^n i = 1 + 2 + \dots + n$$