

# Exercices corrigés sur l'assembleur

## 1 Opérations arithmétiques et logiques

**Exercice 1.1.** Écrivez un code assembleur qui permet de calculer la somme des chiffres de 1 à 9.

**Solution 1.1.**

L'une des solutions pour additionner les chiffres de 1 à 9 pourrait être la suivante:

```
MOV AX,0
MOV BX,1
MOV DX,10
BOUCLE: ADD AX, BX
INC BX
CMP BX, 10
LOOPNE BOUCLE
```

**Exercice 1.2.** Écrivez un code assembleur qui permet de calculer le carré d'un nombre.

**Solution 1.2.**

Le code assembleur pour calculer le carré d'un nombre est comme suit:

```
MOV AX,#nombre
MUL AX
```

**Exercice 1.3.** Écrivez un code assembleur qui permet de calculer la somme des nombres binaires suivants: 1010 et 1001.

**Solution 1.3.**

La somme des deux nombres 1010 (10 en décimal) et 1001 (9 en décimal) pourrait se faire en assembleur de la façon suivante:

```
MOV AX,1010b
MOV BX,1001b
ADD AX,BX
```

Le résultat en binaire sera 10011 et 13 en hexadécimal.

## 2 Sauts conditionnels

**Exercice 2.1.** Écrivez un code assembleur qui permet d'effectuer un saut si la somme de deux nombres est supérieure à 10.

**Solution 2.1.**

Le code assembleur pour effectuer un saut si la somme de deux nombres est supérieure à 10 est le suivant:

```
MOV AX,#nombre1
MOV BX,#nombre2
ADD CX, AX, BX
CMP CX, 10
JG saut
saut: ...
```

**Exercice 2.2.** Écrivez un code assembleur qui permet d'effectuer un saut si un nombre est pair.

**Solution 2.2.**

Pour vérifier si un nombre est pair, nous devons le diviser sur 2 et vérifier le reste de la division. Si le reste est 0 le nombre est pair sinon, il est impair. Le code assembleur pour faire ça est le suivant:

```
MOV AX, #nombre ; le contenu du rY + la valeur 5.
MOV BX, 2
DIV BX ; diviser AX sur 2. Le reste est stocké dans AH.
CMP AH, 0
JE SAUT ; effectuer un saut si le nombre est pair.
... ;reste du code.
```