

**Exercice 1 (6 pts) :**

1. Que représente le jeu d'instructions dans un ordinateur ?
2. Donner puis expliquer les étapes d'exécution d'une instruction pour le mode d'adressage directe ?
3. Dans le cas du processeur 8086, décrire les principaux modes d'adressages et donner un exemple pour chaque mode ?
4. Comment est organisée la mémoire principale dans le cas du processeur 8086 ?

**Exercice 2 (8 pts) :** Soit une carte mère d'un ordinateur contenant les éléments suivants : un processeur de et une mémoire centrale de 256 MO qui est reliée à un bus de données de taille 16 bits. Le tableau suivant résume les instructions utilisées par son microprocesseur :

Instruction	Désignation
ADD X	Acc ← (Acc) + (X)
SUB X	Acc ← (Acc) - (X)
DIV X	Acc ← (Acc) / (X)
MUL X	Acc ← (Acc) * (X)
MOV Acc, X	Acc ← (X)
MOV X, nombre	(X) ← nombre
MOV X, Acc	X ← (Acc)
CMP X, Y	Comparaison de x et y suivi d'une instruction de saut
JE label	Aller au label si x = y
JL label	Aller au label si x < y
JG label	Aller au label si x > y
JMP label	Allez au label sans condition
END	Fin de programme

Où Acc : est le registre accumulateur

(x) : désigne le contenu de la variable x ;

Soit le programme *prog* suivant :

```

Z = A*B + C + D
C = A + D/Z - C
Si Z > C ALORS    A = 2*C
SINON              A = C + Z
FINSI
    
```

1. En utilisant les instructions du tableau, écrire le programme *prog* en assembleur.
2. Calculer la taille d'une adresse mémoire exprimée **en nombre de bits** sachant que la case (emplacement) mémoire est de 1 octet. Justifier votre réponse.

**Exercice 3 (6 pts) :** Soit le programme assembleur ci-dessous :

```

mov ax, 0011
mov bx, FF00
add ax, bx
mov cx, [0011]
mov bx, [FF00]
add cx, bx
    
```

1. Que représente *ax*, *bx*, et *cx* ? Donner l'usage de chacun d'eux.
2. Que fait l'instruction *add ax, bx* ?
3. Quelle est la différence entre les instructions *mov bx, FF00* et *mov bx, [FF00]* ?
4. Pour l'instruction *mov bx, [FF00]*, est-ce que la donnée manipulée est sur 8, 16 ou 32 bits ? Justifier votre réponse.
5. Calculer l'adresse physique (linéaire) générée suite à l'exécution de l'instruction *mov cx, [0011]* sachant que SS contient 0100 et que DS contient 0700. Donner la méthode de calcul.