

## Exercices architecture

Les exercices qui suivent s'appuient sur une structure ultra simplifiée d'un ordinateur totalement imaginaire dont voici les caractéristiques :

La mémoire centrale est constituée d'un tableau de 32 cellules

Chaque cellule est constituée d'un octet

Chaque instruction est constituée

- ❖ d'un code opération sur 3 bits
- ❖ d'une partie adresse sur 5 bits

Le jeu d'instructions est constitué des 8 instructions suivantes :

Instruction	Nom	Effet (version simple)	Effet (version moins simple)	Code opération
LD X	Charger	Charge dans l'accumulateur (l'entrée A de l'UAL) le contenu situé à l'adresse X	$ACC \leftarrow M[X]$	000
STO X	Stocker	Stocke à l'adresse X le contenu de l'accumulateur (le résultat du calcul précédent)	$M[X] \leftarrow ACC$	001
ADD X	Additionner	Ajoute à l'accumulateur (entrée A de l'UAL) le contenu situé à l'adresse X	$ACC \leftarrow ACC + M[X]$	010
SUB X	Soustraire	Soustrait à l'accumulateur (entrée A de l'UAL) le contenu situé à l'adresse X	$ACC \leftarrow ACC - M[X]$	011
JMP ADR	Saut	Saut à l'adresse ADR	$CO \leftarrow A$	100
JMPZ ADR	Saut si = 0	Idem, si résultat du calcul à la ligne d'avant (stocké dans l'accumulateur) est égal à 0	si $CC = 0$ alors $CO \leftarrow A$	101
JMPP ADR	Saut si > 0	Idem, si résultat du calcul dans l'accumulateur est supérieur à 0	si $CC > 0$ alors $CO \leftarrow A$	110
JMPN ADR	Saut si < 0	Idem, si résultat du calcul est inférieur à 0	si $CC < 0$ alors $CO \leftarrow A$	111

On considérera qu'un programme commence à l'adresse 8 (1000 en binaire), et que les adresses 0 à 7 seront utilisées pour stocker les données. On notera END pour la fin de programme

Ex 1.

Que font les programmes ci-dessous ? Version facile en code, version pénible en binaire.

Adresse	Contenu
0	41
1	54
...	
8	LD 0
9	ADD 1
10	STO 2

Adresse	Contenu
0	00100011
1	01000011
...	
1000	00000001
1001	01100000
1010	00100010

Ex 2.

Écrire un programme vérifiant les conditions suivantes :

- ❖ Soit  $x$  une valeur écrite dans la case mémoire 0.
- ❖ Lire le nombre rangé dans la case mémoire 1.
- ❖ Si ce nombre est plus grand ou égal à  $x$ , remplacer le contenu de la case 3 par ce nombre.

Ex 3.

Même exercice que le précédent, mais maintenant, on écrit dans la case mémoire 3 :

- ❖ 0 si le contenu de la case 1 est strictement inférieur à celui de la case mémoire 2
- ❖ 1 sinon.

Ex 4 :

- 1) Écrire en Python un programme qui effectue la multiplication de deux entiers  $a$  et  $b$  positifs ou nuls.
- 2) Écrire le même programme en Assembleur sur les règles précédentes.

Ex 5 : Écrire un programme qui effectue la division euclidienne d'un entier  $a$  par un entier  $b$ . On stockera le quotient et le reste.