

Autotest n°2 – correction

Représentation des données : types et valeurs de base

**EXERCICE 1 :** Revoir les feuilles sur l’algèbre de Boole, les changements de base, l’arithmétique en binaire et les compléments à 2.

**EXERCICE 2 :** Quelle est la valeur en base 10 de  $C6_{16}$ ?

- 1) 36                      2) **198**                      3) 54                      4) 214

**EXERCICE 3 :** Quelle est l’écriture binaire en complément à 2 sur 8 bits de  $-123$ ?

- 1) **10000101**                      2) 0111 1011                      3) 1000 0100                      4) 1000 0110

**EXERCICE 4 :** Quel est le résultat de 
$$\begin{array}{r} 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\ +\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 1 \\ \hline \end{array} ?$$

- 1) 1111 0010                      2) 0001 1100                      3) 1110 0010                      4) **1111 1100**

**EXERCICE 5 :** On considère l’expression booléenne “(a et b) ou (a ou non b)”.

Quelle valeur de a et de b faut-il prendre pour obtenir 0?

- 1)  $a = 0$  et  $b = 0$                       2)  **$a = 0$  et  $b = 1$**                       3)  $a = 1$  et  $b = 0$                       4)  $a = 1$  et  $b = 1$

Représentation des données : types construits

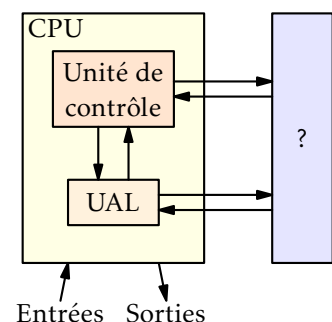
**EXERCICE 6 :** Revoir les exercices de la feuille sur les listes.

Architecture matérielle et systèmes d’exploitation

**EXERCICE 7 :** Revoir la feuille sur les circuits logiques, l’UAL et l’architecture de von Neumann.

**EXERCICE 8 :** Quel est le nom du composant dont il manque le nom dans l’architecture de von Neumann?

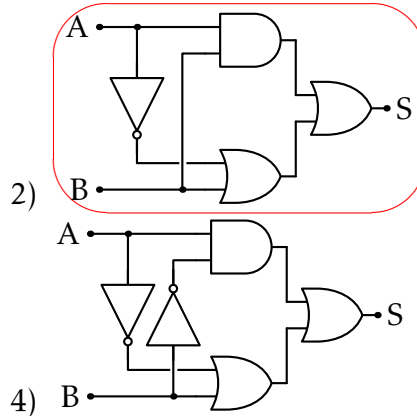
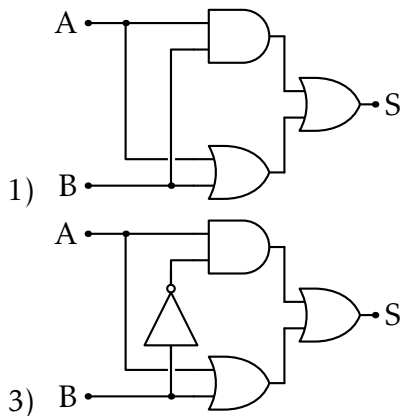
- 1) Le disque dur.  
2) Les registres.  
3) **La mémoire vive.**  
4) L’unité arithmétique et logique.



**EXERCICE 9 :** À quoi servent les registres dans le modèle de von Neumann?

- 1) À faire les calculs arithmétiques ou logiques.  
2) À stocker des valeurs tant que l’ordinateur est allumé.  
3) À stocker des valeurs, même après l’arrêt de l’ordinateur.  
4) **À stocker quelques valeurs et les résultats intermédiaires pour les calculs.**

**EXERCICE 10 :** Quel est le circuit logique correspondant au tableau de vérité ci-contre?



A	B	S
0	0	1
0	1	1
1	0	0
1	1	1

**EXERCICE 11 :** Quelle valeur se trouve dans R0 à la fin de l'exécution de ce programme en assembleur?

- 1) 3      2) 8      3) **9**      4) 12

```
MOV R0, #3
MOV R1, #5
MOV R2, #4
ADD R0, R1, R2
```

**EXERCICE 12 :** Quelle valeur se trouve dans R0 à la fin de l'exécution de ce programme en assembleur?

- 1) 3      2) 6      3) 8      4) **16**

```
MOV R0, #3
MOV R1, #5
CMP R0, R1
BGT saut
ADD R0, R0, R1
saut ADD R0, R0, R0
```

## Algorithmique

**EXERCICE 13 :** On considère la fonction suivante, qui prend deux listes de même longueur.

```
def inconnu(liste1, liste2):
    nouvelle_liste = []
    for i in range(len(liste1)):
        if liste2[i] == 1:
            nouvelle_liste.append(liste1[i])
        # compléter une ligne du tableau
    return nouvelle_liste
```

- 1) Compléter le tableau ci-contre pour l'exécution de `inconnu([3, 1, 9, 0], [1, 1, 0, 1])`.

i	liste1[i]	liste2[i]	nouvelle_liste
			[ ]
0	3	1	[3]
1	1	1	[3, 1]
2	9	0	[3, 1]
3	0	1	[3, 1, 0]

- 2) Quelle est la valeur renvoyée pour `inconnu([1, 1, 0, 1, 0], [1, 0, 0, 1, 1])`?

**Solution :** On obtient [1, 1, 0].

3) Décrire en français ce que fait cette fonction. Par exemple : “elle fait la somme de tous les éléments de la liste”.

**Solution :** La fonction renvoie une nouvelle liste qui ne contient que les éléments de liste1 tels que l'élément de liste2 de même indice vaut 1.