Lycée Les 3 Sources — **Bourg-Lès-Valence**
NSI 1re — Année **2021-22**

Devoir surveillé n°2 – correction

Les exercices 1 à 11 sont des QCM. Pour chaque question entourer la seule réponse est correcte

Représentation des données : types et valeurs de base

Exercice 1 : Quelle est la valeur en hexadécimal de 107_{10} ?

1) 611 2) A7 3) 6B 4) 6A

Exercice 2 : Quelle est l'écriture binaire en complément à 2 sur 8 bits de −74?

1) 01001010
2) 10110110
3) 10110101
4) 10110111

128	64	32	16	8	4	2	1

Exercice 3 : Quel est le résultat de $\begin{array}{r} 01000111 \\ +\,01101011 \\ \hline \end{array}$?

1) 00101100 2) 10110000 3) 10110010 4) 11100010

Exercice 4 : Quelle est la table de vérité de l'expression E = (non(a) et b) ou non(a et b)?

1)

a	b	E
0	0	1
0	1	1
1	0	1
1	1	0

2)

a	b	E
0	0	1
0	1	1
1	0	0
1	1	1

3)

a	b	E
0	0	0
0	1	1
1	0	1
1	1	0

4)

a	b	E
0	0	1
0	1	1
1	0	1
1	1	1

Exercice 5 : On considère l'expression booléenne :

(a et c) ou ((b et c) ou (non (b) et non(c)))

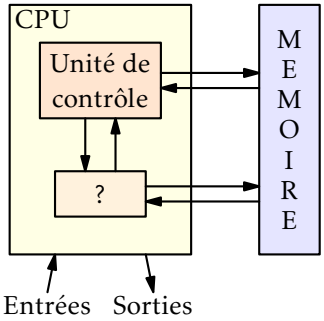
Quelles valeurs de a, b et c permettent d'obtenir 0?

1) $a = 0, b = 0, c = 0$ 2) $a = 0, b = 1, c = 1$ 3) $a = 1, b = 0, c = 1$ 4) $a = 1, b = 1, c = 0$

Architecture matérielle et systèmes d'exploitation

Exercice 6 : Quel est le nom du composant dont il manque le nom dans l'architecture de von Neumann?

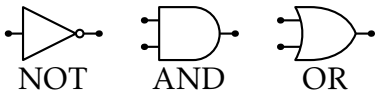
1) La carte graphique.
2) Les registres.
3) L'alimentation.
4) L'unité arithmétique et logique.

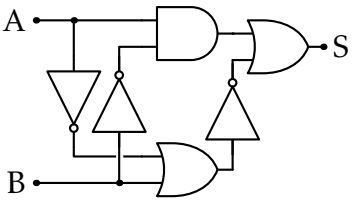


Exercice 7 : À quoi sert la mémoire vive (RAM) dans le modèle de von Neumann ?

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 afficher les valeurs à l'écran.

Exercice 8 : On considère le circuit ci-contre. Quelle est la table de vérité de ce circuit ?
Les portes utilisées sont :





1)

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

2)

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

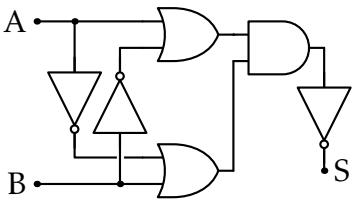
3)

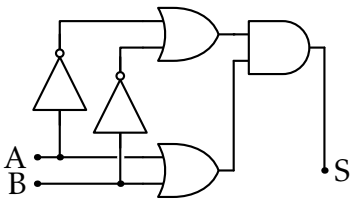
A	B	S
0	0	0
0	1	1
1	0	0
1	1	0

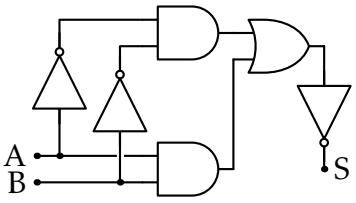
4)

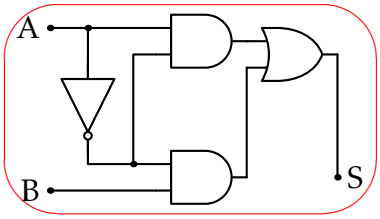
A	B	S
0	0	1
0	1	1
1	0	1
1	1	1

Exercice 9 : Quel est le circuit logique qui n'a pas la même table de vérité que les autres ?

1)


2)


3)


4)


Exercice 10 : Quelle valeur se trouve dans R2 à la fin de l'exécution de ce programme en assembleur ?

1) 3 2) 5 3) 7 4) 13

Assembleur	Signification
`MOV dest, op1`	dest = op1
`ADD dest, op1, op2`	dest = op1 + op2

```
        MOV     R0, #1
        MOV     R1, #2
        MOV     R2, #3
        ADD     R0, R1, R2
        ADD     R2, R1, R0
        END
```

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

1) 5 2) 10 3) 12 4) 15

Assembleur	Signification
`CMP op1, op2` `BLT label`	Aller à l'instruction `label` si op1 < op2

```
        MOV     R0, #0
        MOV     R1, #5
boucle  ADD     R0, R0, R1
        CMP     R0, #12
        BLT     boucle
        END
```

Représentation des données : types construits

Exercice 12 : (2pt) On considère les instructions suivantes :

```
>>> liste1 = [3, 7, 1]
>>> liste2 = [5, 6]
>>> liste3 = [0, 9, 4]
```

1) Que vaut chacune des expressions suivantes :
   a) `liste1[1]` : 7
   b) `liste3 + liste2` : `[0, 9, 4, 5, 6]`
   c) `liste2[1] + liste1[2]` : 7 (on fait la somme de 6 et de 1)
2) On rajoute les instructions suivantes :

```
>>> liste2.append(5)
>>> liste2.append(liste1[0])
```

   Quelle est alors la valeur de `liste2` ? `[5, 6, 5, 3]`

Exercice 13 : (2pt) Déterminer les listes obtenues à l'aide des expressions suivantes :

1) `[2*i for i in range(5)]` : `[0, 2, 4, 6, 8]` (on multiplie par 2 chaque élément de `[0, 1, 2, 3, 4]`)
2) `[i for i in range(10) if i >= 5]` : `[5, 6, 7, 8, 9]` (on ne garde que les éléments supérieurs ou égaux à 5)

Algorithmique

Exercice 14 : (4pt) On considère la fonction ci-dessous.

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

1) Compléter le tableau ci-contre pour l'exécution de `inconnu([5, 4, 1, 7], [2, 8, 2, 1])`.

i	liste1[i]	liste2[i]	nouvelle_liste
			[]
0	5	2	[5]
1	4	8	[5, 8]
2	1	2	[5, 8, 2]
3	7	1	[5, 8, 2, 7]

2) Quelle est la valeur renvoyée pour `inconnu([3, 1, 6], [9, 2, 5])` ? `[9, 2, 6]`

3) Décrire en français ce que fait cette fonction. Par exemple : "elle fait la somme de tous les éléments de la liste". On suppose que `liste1` et `liste2` ont la même longueur.
Cette fonction fait une liste avec à chaque fois le plus grand des deux éléments de `liste1` et `liste2` de même indice. S'ils sont égaux, rien n'est rajouté.

4) Quelle est la valeur renvoyée pour `inconnu([5, 7, 3], [2, 7, 6])` ? `[5, 6]`

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

Exercice 15 : (2pt) Écrire le code de la fonction `maximum(liste)` qui renvoie la plus grande valeur contenue dans `liste`. On suppose que la liste n'est pas vide. Vous ne pouvez pas utiliser la fonction `max(liste)`.

```
def maximum(liste):
    m = liste[0]
    for v in liste:
        if v > m:
            m = v
    return m
```

```
>>> maximum([5, 1, 8, 14, 2])
14
>>> maximum([15])
15
```

```
>>> maximum([-9, -27, -2, -48])
-2
>>> maximum([9, 2, -57, 1, 8])
9
```