

DS01 Thème 1: Bases de la programmation Thème 2: Binaire – Circuit – Codage	- 1_SPE NSI3	Le 12/11/2020 Durée: 1h30
Calculatrice non autorisée Aucun document autorisé Réponses sur feuilles doubles numérotées		

Exercice 1: (10 points)

- 1.1. Donner l'écriture binaire du nombre dont l'écriture décimale est 2
- 1.2. Donner l'écriture décimale du nombre dont l'écriture binaire est 1 0101
- 1.3. Donner l'écriture hexadécimal du nombre dont l'écriture décimale est 20
- 1.4. Etablir la table de vérité de la fonction $Not(x) And Not(y)$
- 1.5. Programmer une instruction qui affiche un texte entré par l'utilisateur depuis le clavier
- 1.6. Donner la valeur des variables après exécution du programme ci-dessous:

```
a = 3
a = a + 1
b = a
b = b * 2
c = a ** b
```

- 1.7. Ecrire ce qui sera affiché après exécution du programme ci-dessous:

```
a = 7
if a % 2 == 0 :
    print(a)
elif True :
    print("OK")
else :
    print(« KO" )
```

- 1.8. Décrire le traitement réalisé par la fonction ci-dessous (a et b sont des nombres entiers):

```
def maFonction( a , b ):
    c = 1
    for i in range( a ):
        c = c * b
    return c
```

- 1.9. Décrire ce qui se passera après appel de la fonction ci-dessous (a est un nombre entier):

```
def maFonction( a ):
    i = 0
    while i < a :
        print( "NSI" )
```

- 1.10. Donner la valeur des variables c et d après exécution du programme ci-dessous:

```
a = [ 1 , 2 , 3 , 4 , 5 ]
b = [ a , a , a ]
c = a [ 0 ]
d = b [ 1 ]
```

Exercice 2: (7 points)

Programmer une fonction qui prend une liste en paramètre et renvoie une liste contenant les mêmes éléments mais en sens inverse:

```
a = inverseListe ([ 1 , 2 , 3 , 4 ])          # a = [ 4 , 3 , 2 , 1 ]
```

```
b = inverseListe ( [ "NSI" , True , 0.1 , 5 ] )    # b = [ 5 , 0.1 , True , "NSI" ]
```

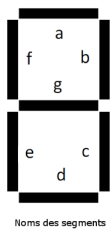
Vous ne pouvez pas utiliser de fonction native.

Vous pouvez utiliser la méthode append qui ajoute un élément en fin de liste:

```
c = [ 1 , 2 , 3 ]
c.append ( 4 )          # c = [ 1 , 2 , 3 , 4 ]
```

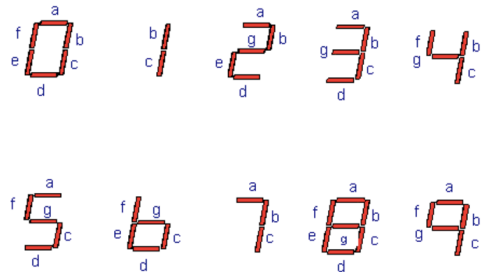
Exercice 3: (7 points)

Un afficheur 7 segments permet d'afficher un chiffre décimal (de 0 à 9) en allumant les segments correspondants.



On dispose d'une machine qui allume les segments correspondant au nombre décimal codé par 4 entrées $b_3 b_2 b_1 b_0$ représentant son codage binaire.

Par exemple, si on active les entrées du codage binaire du nombre décimal 3, les segments a, b, c et d seront activées en sortie.



Combinaisons de segments permettant d'obtenir les chiffres de 0 à 9.

Etablir la table de vérité de la sortie a

Exercice 4: (6 points)

Vous réalisez en binôme un projet sous Python avec le module Turtle où vous devez dessiner un pavage du plan d'un motif obtenu par rotation d'une forme de base.

Votre binôme vous a livré la fonction motif (tailleMotif) qui:

- prend en paramètre tailleMotif : la taille du motif (int)
- dessine un motif inscrit dans un carré de côté de longueur la taille du motif donnée en paramètre
- commence et termine le motif au coin inférieur gauche de ce carré

Vous devez l'utiliser pour programmer la fonction pavage (tailleMotif , taillePavage) qui:

- prend en paramètres:
 - tailleMotif : la taille du motif (int)
 - taillePavage : le nombre de lignes et le nombre de colonnes du pavage (list)
- dessine un pavage du motif avec le nombre de lignes et de colonnes données en paramètres
- n'utilise pas la fonction goto () de turtle mais uniquement les fonctions turtle:
 - forward (x) # avance de x pixels
 - right (y) # tourne de y degrés vers la droite
 - penup () # pour lever le stylo
 - pendown () # pour descendre le stylo

Programmer la fonction pavage ()