

## Devoir maison N°2

### Exercice 1

Voici un algorithme :

Données : **liste** (une liste composée de 1 et de 0 )  
 Résultat : **nouvelle\_liste** (une autre liste constituée de 1 et de 0 )

```
nouvelle_liste prend la valeur [ ]
pour i parcourant les positions de liste faire
    si liste[i] vaut 1 alors
        nouvelle_liste se voit ajouter 0
    sinon
        nouvelle_liste se voit ajouter 1
    fin si
fin pour
```

**Q1. Traduire** cet algorithme et le **transformer** en fonction Python sur la feuille.

**Q2. Expliquer** en 2 lignes maximum ce que fait cet algorithme.

### Exercice 2

Le code ci-contre définit une fonction qui permet de dessiner des polygones. Mais le code a été mélangé et les différentes indentations ont été complètement perdues.

**Q1. Indiquer** le nom de cette fonction ainsi que le nombre et le(s) noms des paramètre(s).

**Q2. Ordonner** ce code afin qu'il dessine un polygone et **l'écrire** sur feuille.

**Q3. Décrire** les modification à apporter pour que cette fonction dessine un pentagone de 50 pixels de côté.

**Q4. Décrire** les résultats des instructions **polygone(4, 100)** et **polygone(100, 4)**

**Q5.** A partir de cette fonction, proposer deux fonctions capables de dessiner les deux figures suivantes.

```
forward(longueur_cote)
from turtle import *
for i in
range(nb_etapes):
def polygone(nb_etapes,
longueur_cote):
left(360/nb_etapes)
```

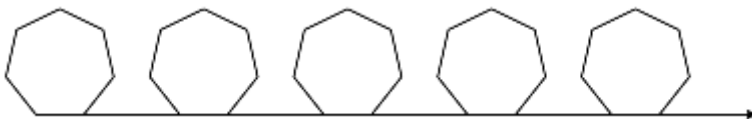


Figure 2: Première figure attendue

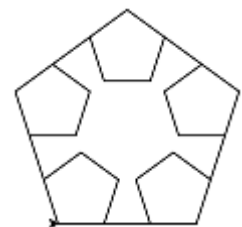


Figure 1: Seconde figure attendue

## Exercice 3 – Dentiste

Chez le dentiste, la bouche grande ouverte, lorsqu'on essaie de parler, il ne reste que les voyelles. Nous allons écrire une fonction **dentiste(texte)** qui renvoie un texte ne contenant que les voyelles de texte, dans l'ordre. Les voyelles sont a, e, i, o, u et y. Même les ponctuations sont supprimées.

Exemple :

```
>>> dentiste("j'ai mal")  
'aia'
```

**Q1. Donner** le résultat de l'instruction **dentiste("il fait chaud")**

Pour écrire la fonction **dentiste(texte)**, on pourra s'inspirer ou utiliser les fonctions :

→ **recopier(texte)** qui prend un texte et renvoie une copie de ce texte. On part du texte vide et on parcourt toutes les lettres de texte en les rajoutant à droite de résultat. Cette fonction s'écrit :

```
def recopier(texte):  
    resultat = ""  
    for lettre in texte:  
        resultat = resultat + lettre  
    return resultat
```

→ **voy(lettre)** qui renvoie un booléen indiquant si lettre est une voyelle comme indiqué sur les test suivants.

```
>>> voy('a')  
True  
>>> voy('b')  
False
```

**Q2. Ecrire** en langage Python la fonction **dentiste(message)**

## Exercice 4 : Fonction mystère

On considère le code suivant:

```
def mystere1(tab):  
    n = len(tab)  
    s = ''  
    for i in range(n):  
        if tab[i] > i:  
            s = s + "+" + str(tab[i])  
    return s
```

```
def mystere2(tab):  
    i = 0  
    s = 0  
    c = 0  
    while tab[i] < 4:  
        s = s + tab[i]  
        c = c + s  
        i = i + 1  
    return (s,c)
```



**Q1. Donner et justifier** les résultats des instructions suivantes :

```
mystere1( [ 1 , 0 , 2 , 4 , 3 ] )
```

```
mystere2( [ 1 , 0 , 2 , 4 , 3 ] )
```

## Exercice 5 – fonctions élémentaires (à faire sur ordinateur)

**Q1. Définir** une fonction **nombre\_positifs(t)** qui renvoie le nombre d'éléments positifs dans le tableau **t**.

L'exécution devra donner :

```
>>> nombre_positifs( [ 7 , -2 , 5 , 6 , -10] )
3
```

**Q2. Définir** une fonction **compte\_lettre(l , ch)** qui compte le nombre de fois que la lettre **l** apparaît dans la chaîne **ch**:

```
>>> compte_lettres("a", "blablacar")
3
>>> compte_lettres("b", "blablacar")
2
```

**Q3. Définir** une fonction **fois\_dix(t)** qui prend en argument un tableau de nombres et renvoie un tableau constitué des 3 derniers nombres multipliés par 10.

```
>>> fois_10([ 5 , 2 , 3 , 8])
[ 20 , 30 , 80]
```

