

Devoir d'informatique

Exercice 1 – Structure If

Question 1 Ecrire la fonction $f(x:\text{int}) \rightarrow \text{int}$ qui permet d'implémenter la fonction f qui à x associe

$$\begin{cases} 2 & \text{si } x \in]-\infty, -2[\\ -x & \text{si } x \in [-2, 0] \\ 0 & \text{si } x \in [0, +\infty[\end{cases} .$$

Correction

```
def f(x):
    if x < -2 :
        return 2
    elif x <= 0 :
        return -x
    else :
        return 0
```

Exercice 2 – Structure For

Question 2 Ecrire une fonction $\text{affiche}(n:\text{int}) \rightarrow \text{None}$ affichant les entiers multiples de 5 compris entre 1 inclus et n inclus. On utilisera une boucle for.

Correction

```
def affiche(n):
    for i in range(1, n+1) :
        if i%5 == 0 :
            print(i)
```

Exercice 3 – Structure While

Question 3 Ecrire une fonction $\text{affichew}(n:\text{int}) \rightarrow \text{None}$ affichant les entiers multiples de 5 compris entre 1 inclus et n inclus. On utilisera une boucle while.

Correction

```
def affichew(n):
    i = 1
    while i <= n :
        if i%5 == 0 :
            print(i)
```

```
i=i+1
```

Exercice 4 – Structure For ou While

Question 4 Ecrire une fonction `prod(n:int) -> int` calculant le produit des entiers impairs de 0 à n (exclus).

Correction

```
def prod(n):
    res = 1
    for i in range(0,n):
        if i%2 != 0 :
            res = res*i
    return res
```

Exercice 5 – Implémentation d'une suite

On pose $u_0 = 1$ et $u_1 = 2$. Pour tout $n \in \mathbb{N}$ strictement supérieur à 1, $u_{n+2} = 2u_{n+1} - u_n$.

Question 5 Ecrire une fonction `u(n:int)->n` permettant de renvoyer le nième terme de la suite u .

Correction

```
def u(n):
    u,v = 1,2
    for i in range(2,n):
        u,v = v,(2*v-u)
    return u
```

Exercice 6 – Analyse d'ue fonction

Soit la fonction suivante.

```
def sqrt_int(n):
    """Renvoie la partie entière de la racine carrée de n"""
    s = 0
    while s**2 <= n:
        s = s+1
        s = s-1
    return s
```

Question 6 Donner l'évolution de la variable s lors de l'appel suivant : `sqrt_int(2)`. On se limitera à 4 itérations. .

Correction Boucle infinie, s n'évolue jamais.

Question 7 Corriger la fonction pour pour qu'elle coïncide avec le but annoncé.

Correction

```
def sqrt_int(n):
    """Renvoie la partie entière de la racine carrée de n"""
    s = 0
    while (s+1)**2 <=n:
        s = s+1
    return s
```

Exercice 7 – Quelques questions sur les chaînes de caractères

Question 8 Ecrire une fonction `check(mot:str, lettre:str) -> bool` qui renvoie True si la lettre

lettre est dans le mot mot, False sinon.

Correction

```
def check(mot:str, lettre:str) -> bool :  
    return lettre in mot
```

Question 9 Ecrire une fonction `compte(mot:str, lettre:str) -> int` qui renvoie le nombre d'occurrences de la lettre `lettre` dans le mot `mot`.

Correction

```
def compte(mot:str, lettre:str) -> int :  
    cpt = 0  
    for l in mot :  
        if l == lettre :  
            cpt = cpt+1  
    return cpt
```

Question 10 Ecrire une fonction `proportion(mot:str, lettre:str) -> float` qui donne la proportion `lettre` dans le mot `mot`.

Correction

```
def proportion(mot:str, lettre:str) :  
    return compte(mot, lettre)/len(mot)
```

Question 11 Ecrire une fonction `remplace(mot:str, lettre1:str, lettre2:str) -> str` qui remplace la lettre `lettre1` par la lettre `lettre2` dans le mot `mot`.

Correction

```
def replace(mot:str, lettre1:str, lettre2:str) :  
    res = ''  
    for lettre in mot :  
        if lettre == lettre1 :  
            res = res + lettre2  
        else :  
            res = res + lettre  
    return res
```