

Devoir-Surveillé - Informatique

CORRECTION

Exercice 0.1. *Ecrire un programme sous Python qui :*

- demande un entier naturel non nul n à l'utilisateur ;
- enregistre n nombres décimaux dans une liste L ;
- affiche la liste P des indices des valeurs strictement positives de la liste L .

Exemple : Si $L = [-5, 3, 2, -4, 1]$ alors $P = [1, 2, 4]$.

```
• n=int(input('saisir le nombre de valeurs'))
• L=[]
• P=[]
• for i in range(n):
•     x=float(input('saisir une valeur'))
•     L.append(x)
•     if x>0:
•         P.append(i)
• print(P)
```

Exercice 0.2. *En 1640, Pierre de Fermat conjectura, dans plusieurs correspondances avec divers mathématiciens, que les nombres $F_n = 2^{2^n} + 1$, où n est un entier naturel, sont tous premiers. Ces nombres furent nommés par la suite « nombres de Fermat ».*

1. *Ecrire un programme sous Python qui :*

- demande à l'utilisateur de saisir un entier naturel p ;
- affiche *True* si p est premier et *False* sinon.

On rappelle qu'un nombre entier $n > 2$ est premier s'il n'admet aucun diviseur dans $\llbracket 2; n-1 \rrbracket$.

```
• p=int(input('saisir un entier'))
• test = True # on initialise une variable test à True
• for d in range(2,p-1):
•     if p%d==0: # dès qu'on rencontre un diviseur entre 2 et p-1...
•         test = False # on change la variable test en False
• print(test)
```

*On suppose pour la suite qu'il existe une fonction **premier(p)** qui prend comme argument d'entrée un entier p et qui renvoie **True** si p est premier, et **False** sinon. Et on pourra l'utiliser pour la suite de l'exercice, en écrivant par exemple **premier(11)**, instruction pour laquelle Python renverra **True**.*

2. Écrire un programme demandant à l'utilisateur de saisir $n \in \mathbb{N}$ et qui affiche :

- la valeur de F_n
- si F_n est premier ou pas

```
def premier(p):
    • test = True # on initialise une variable test a True
    • for d in range(2,p):
    •     if p%d==0: # des qu'on rencontre un diviseur entre 2 et p-1...
    •         test = False # on change la variable test en False
    • return(test)

• n=int(input('saisir un entier'))
• cpt=0 # initialisation du compteur
• F=2**(2**n)+1
• print(F, '_', n, '=', F)
• if premier(F):
•     print(F, '_', n, 'est premier')
• else :
•     print(F, '_', n, 'n est pas premier')
```

3. Écrire un programme déterminant le premier entier naturel n tel que F_n n'est pas premier.

```
• n=0
• while premier(2**(2**n)+1):
•     n=n+1
• print(n)
```

Remarque : Ce fut Leonhard Euler qui, en 1732, réfuta cette conjecture en prouvant que $F_5 = 4294967297$ est multiple de 641. Par la suite, on a constaté que les nombres de Fermat pour $n > 5$ semblent n'être en fait... jamais premiers. Mais là encore, c'est une conjecture : en 2013, on avait vérifié que pour tout $n \in \llbracket 5; 2747497 \rrbracket$, F_n n'est pas premier, mais on ne sait toujours pas s'il existe des nombres de Fermat premiers pour n plus grand.

Exercice 0.3. Soit la suite (u_n) définie par la relation de récurrence $f(x) = \begin{cases} u_1 = 8 \\ u_{n+1} = \sqrt{5u_n - 4} \end{cases}$

1. Une fonction **terme**(N) prenant pour paramètre d'entrée un entier N et qui renvoie u_N .
2. Une fonction **somme**(N) prenant pour paramètre d'entrée en entier N et qui renvoie $S_N = \sum_{k=1}^N u_k$.

3. La suite (u_n) converge vers 4.

Écrire une fonction **rang(arrondi)** prenant pour paramètre d'entrée un flottant nommé arrondi et qui renvoie le plus petit entier n tel que $|u_n - 4| < \text{arrondi}$.

```
def terme(n):
    #reçoit le rang N et renvoie uN
    s=8
    for i in range(1,n):
        s=sqrt(5*s-4)
    return(s)

def somme(n):
    s=0
    for i in range(1,n+1):
        s=s+terme(i)
    return(s)

def rang(arrondi):
    n=1
    while terme(n)-4>arrondi:n=n+1
    return(n)
```

Exercice 0.4. Écrire un programme sous Python qui demande à l'utilisateur de saisir un texte puis qui affiche ce texte en remplaçant les 'i' par des 'o' et les 'o' par des 'i'.

```
• texte=str(input('saisir un texte'))
• nouv='' #texte vide
• for i in range(len(texte)):
•     if texte[i]=='i':
•         nouv=nouv+'o' # le i est remplacé par un o
•     elif texte[i]=='o':
•         nouv=nouv+'i' # le o est remplacé par un i
•     else:
•         nouv=nouv+texte[i]
• print('le nouveau texte est : ',nouv)
```

Exercice 0.5. Trouver en justifiant le nombre à virgule représenté par le mot de 32 bits :

00101110101011100000000000000000 .

- Le premier bit est 0 donc le flottant codé est positif.
- $e_C = 01011101_{(2)} = 64_{(10)} + 16_{(10)} + 8_{(10)} + 4_{(10)} + 1_{(10)} = 93_{(10)}$
Donc $e_R = e_C - (2^{8-1} - 1) = 93 - 127 = -34$.
- La mantisse est 010111000000000000000000, soit : $m = 2^{-2} + 2^{-4} + 2^{-5} + 2^{-6}$.

Ainsi le nombre codé est : $2^{-34}(1 + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-6})$