T
experts

Interrogation écrite du mardi 9 janvier 2024

Durée : 20 minutes

Numéro : ….. **Prénom et nom :** ……. **Note : ….. / 20**

- L'usage de la calculatrice n'est pas autorisé ; celui d'une fiche préparée à l'avance non plus.
- Écrire très lisiblement et sans ratures.

I. (8 points : 1°) 2 points + 2 points ; 2°) 4 points)

1°) La fonction écrite dans le cadre ci-dessous prend pour argument un entier naturel *x* et renvoie la somme des chiffres de son écriture en base dix.

```
def somme_chiffres(x):
    s = 0
    while x!=0:
        s = s + x % 10
        x = x//10
    return s
```

Répondre par oui ou non aux deux questions suivantes :

- Peut-on échanger les instructions `s = s + x % 10` et `x = x//10` ? …….
- Peut-on remplacer la condition `x!=0` par `x>0` ? …….

2°) En utilisant la fonction du 1°), écrire dans le cadre ci-dessous une fonction Python d'en-tête `def s(n):` qui prend pour argument un entier naturel *n* et qui renvoie tous les entiers naturels inférieurs ou égaux à *n* dont la somme des chiffres de leur carré en base dix est égale à 9.
On utilisera une boucle `for`.

……

……

……

……

II. (8 points : 3 points + 5 points)

Le but de l'exercice est d'écrire une fonction Python d'en-tête `def pairs(L) :` qui prend en argument une liste L d'entiers relatifs et qui renvoie la liste M, éventuellement vide, de tous les éléments pairs de L à la même place.

On envisage deux façons différentes.

1ère façon :

Écrire le programme en 2 lignes, en-tête compris, en utilisant une instruction du type :
`return [x for x in L if .....]`.

```
    ……..…
    ……..…
```

2e façon :

Écrire le programme en 6 lignes, en-tête compris, en utilisant une boucle `for` et la fonction `append`.
On utilisera également une liste M initialement vide.

```
    ……..…
    ……..…
    ……..…
    ……..…
    ……..…
    ……..…
```

III. (4 points : 2 points + 2 points)

La fonction Python d'en-tête `def test_pair(L) :` écrite dans le cadre ci-dessous prend en argument une liste L d'entiers relatifs et a pour but de renvoyer la valeur `True` si la liste contient au moins un entier pair et la valeur `False` sinon. Compléter les pointillés après le `if` et le `return`.

```
def test_pair(L) :
    c=False
    for x in L :
       if ……..…. :
          c=True
    return ….
```

Corrigé de l'interrogation écrite du 9-1-2024

I.

1°) La fonction écrite dans le cadre ci-dessous prend pour argument un entier naturel *x* et renvoie la somme des chiffres de son écriture en base dix.

```
def somme_chiffres(x):
    s = 0
    while x!=0:
          s = s + x % 10
          x = x//10
    return s
```

Répondre par oui ou non aux deux questions suivantes :

• Peut-on échanger les instructions `s = s + x % 10` et `x = x//10` ? non

• Peut-on remplacer la condition `x!=0` par `x>0` ? oui

2°) En utilisant la fonction du 1°), écrire dans le cadre ci-dessous une fonction Python d'en-tête `def s(n):` qui prend pour argument un entier naturel *n* et qui renvoie tous les entiers naturels inférieurs ou égaux à *n* dont la somme des chiffres de leur carré en base dix est égale à 9.
On utilisera une boucle `for`.

```
def s(n):
    for x in range(n+1):
      if somme_chiffres(x**2)==9:
         print (x)
```

Exemple :
$9^2 = 81$
La somme des chiffres de 81 est égale à 9.

II.

Le but de l'exercice est d'écrire une fonction Python d'en-tête `def pairs(L) :` qui prend en argument une liste L d'entiers relatifs et qui renvoie la liste M, éventuellement vide, de tous les éléments pairs de L à la même place.

On envisage deux façons différentes.

1ère façon :

Écrire le programme en 2 lignes, en-tête compris, en utilisant une instruction du type :
`return [x for x in L if .....]`.

```
def pairs(L) :
    return [x for x in L if x%2==0]
```

2e façon :

Écrire le programme en 6 lignes, en-tête compris, en utilisant une boucle `for` et la fonction `append`.
On utilisera également une liste M initialement vide.

```
def pairs(L) :
    M=[]
    for x in L:
        if x%2==0
           M.append(x)
    return M
```

Il y a d'autres possibilités comme celle écrite dans l'encadré ci-dessous.

```
def pairs(L) :
    M=[]
    for i in range(len(L)):
        if L[i]%2==0:
           M.append(L[i])
    return M
```

III.

La fonction Python d'en-tête `def test_pair(L) :` écrite dans le cadre ci-dessous prend en argument une liste L d'entiers relatifs et a pour but de renvoyer la valeur `True` si la liste contient au moins un entier pair et la valeur `False` sinon. Compléter les pointillés après le `if` et le `return`.

```
def test_pair(L) :
    c=False
    for x in L :
       if x%2==0:
          c=True
    return c
```

Il s’agit d’une fonction booléenne (renvoie une valeur de vérité).