

Contrôle Continu 1 - Épreuve Écrite

Durée 90 min

15 Octobre 2018

*L'usage de tout appareil électronique, ouvrage de référence ou notes de cours est rigoureusement interdit.
Les exercices peuvent être traités dans l'ordre que vous souhaitez.*

Toute fonction définie au cours de l'examen peut être réutilisée dans les questions suivant sa déclaration

Exercice 1 - Questions de cours

Qu'affichent les programmes suivants ? Indiquer simplement **Erreur** si le programme contient une erreur. Chaque programme est indépendant.

1.

```
l = [1,2,3]
for x in l:
    x *= 2
print(l)
```

2.

```
l = [1,2,3]
m = 1
m[1] += 1
print((1,m))
```

3.

```
print("Nous sommes le "+15+"octobre")
```

4.

```
print(2+4*-3+-2**2*4)
```

5.

```
x = 4
if x % 2 < 1:
    print("pair")
else:
    print("négatif et impair")
```

6.

```
x = 3
y = 6
if x * y = 18:
    print("un")
if x - y > 0:
    print("deux")
if x < y:
    print("trois")
```

7.

```
l = []
for i in range(2, 10):
    l.append("")
print(len(l))
```

Tourner la page SVP.

Exercice 2 - Somme des termes d'une suite récurrente

On considère la suite (u_n) définie par la relation de récurrence suivante avec $u_0 = 1$ et

$$u_{n+1} = \sqrt{u_n} + \frac{1}{n+1} \quad \text{pour } n \geq 1. \quad (1)$$

(Q_{1.}) Écrire un programme ou une fonction `python` qui calcule la somme des n premiers termes de la suite (u_n) .

Exercice 3 - Maximum à quatre arguments

On considère la fonction suivante

$$\text{max4}(x_1, x_2, x_3, x_4) \quad (2)$$

qui renvoie la plus grande valeur parmi les grandeurs réelles x_1, x_2, x_3, x_4 .

(Q_{1.}) Écrire un programme ou une fonction `python` qui trouve le maximum parmi quatre grandeurs réelles `x1`, `x2`, `x3`, `x4`.

Remarque. Attention ! Toute utilisation de la fonction `max` appartenant à la librairie standard `Python` est interdite.

Exercice 4 - Liste de nombres premiers

L'objectif de cet exercice est d'obtenir la liste des nombres premiers plus petits qu'un entier $N \in \mathbb{N}$ donné. Chaque question peut être traitée de manière indépendante. Il est toutefois conseillé de les traiter dans l'ordre indiqué.

Définition. Un entier p est dit premier, si $p \geq 2$ et aucun entier compris entre 2 et $p - 1$ ne divise p .

$$p \text{ est premier} \Leftrightarrow [p \geq 2 \quad \text{et} \quad \forall m \in \{2, \dots, p-1\}, \quad p \bmod m \neq 0]$$

(Q_{1.}) Écrire un programme ou une fonction qui teste si un entier `p` est premier ou non. On vous demande ici d'utiliser une boucle `for`.

(Q_{2.}) Réécrire le programme ou la fonction précédent(e) en utilisant à présent une boucle `while`. La boucle `while` s'arrêtera à la première contradiction sur la primalité de `p`.

(Q_{3.}) Écrire un programme ou une fonction qui à partir d'un entier N génère une liste `lst`, telle que pour tout $k \leq N$ `lst[k] = True` si k est premier et `lst[k] = False` sinon.

(Q_{4.}) Écrire un programme ou une fonction qui génère la liste des premiers entiers inférieurs ou égaux à N .

Définition. Le crible d'Ératosthène est une méthode permettant d'assigner à tous les entiers inférieurs à $N \in \mathbb{N}$ la valeur `True` s'ils sont premiers, ou `False` s'ils ne sont pas premiers.

On peut résumer l'algorithme comme suit :

1. Initialiser une liste de taille $N+1$ qui contient `False` en 0 et en 1, et `True` partout ailleurs.
2. Pour chaque entier k compris entre 2 et $\lfloor \sqrt{N} \rfloor$:
 - Si la valeur assignée à k est `True`, alors assigner `False` à tous les multiples $m \leq N$ de k .
 - Si la valeur assignée à k est `False`, ne rien faire.

Remarque. L'opérateur $\lfloor \cdot \rfloor$ désigne la partie entière.

(Q_{5.}) Écrire un programme ou une fonction implémentant le crible d'Ératosthène.