

Nom :

Prénom :

Matricule :

MATH2010-1 Logiciels mathématiques Examen final écrit – 16 juin 2016

Répondre aux questions suivantes. Vos réponses doivent être manuscrites (rédigées à la main) et utiliser le langage Python 3. Vous avez le droit d'utiliser un crayon et rien d'autre.

Question 1 (5 pts). Que sera-t-il imprimé à l'écran par l'exécution du code suivant ?

```
L = []
for i in range(10):
    if i % 2 == 0:
        L.append(i)
    elif i % 3 == 0:
        L.append(i*10)
    elif i == 6:
        L.append(i*100)
    if i == 8:
        L.append(i*1000)
print(L)
```

Réponse :

[0, 2, 30, 4, 6, 8, 8000, 90]

Question 2 (5 pts). En supposant que `isprime(p)` retourne `True` si `p` est premier et `False` sinon, trouver trois façons d'améliorer le code suivant :

```
def triplets_nombres_premiers(n):
    L = []
    p = 3
    while len(L) < n:
        if isprime(p) == True and isprime(p+6) == True and isprime(p+2) == True:
            L.append((p, p+2, p+6))
        elif isprime(p) == True and isprime(p+6) == True and isprime(p+4) == True:
            L.append((p, p+4, p+6))
        p += 1
    return L
```

Façon 1 : Enlever les `== True`.

Façon 2 : Tester une seule fois `isprime(p)` and `isprime(p+6)`.

Façon 3 : Remplacer `p += 1` par `p += 2`.

Question 3 (3 pts). Trouver un nom adéquat pour la fonction suivante. Écrire une courte phrase entre la paire de triple-guillemets qui décrit ce que la fonction fait.

```
def _____(n):  
    L = []  
    for i in range(1, n):  
        if n % i == 0:  
            L.append(i)  
    return L
```

Nom de la fonction : `diviseurs_propres`

Description : `Retourne la liste des diviseurs propres de n.`

Question 4 (7 pts). Un nombre est dit *parfait* s'il est égal à la somme de ses diviseurs propres. Par exemple, le nombre 6 est parfait, car les diviseurs propres de 6 sont 1, 2 et 3 et $6 = 1 + 2 + 3$. Le nombre 15 n'est pas parfait, car ses diviseurs propres sont 1, 3 et 5 et $15 \neq 1 + 3 + 5$. Écrivez une fonction Python appelée `nombre_parfait(n)` qui retourne la liste des nombres parfaits strictement inférieurs à `n`. Vous pouvez structurer votre programme en définissant d'autres fonctions si nécessaire.

Réponse :

```
def nombre_parfaits(n):  
    L = []  
    for i in range(1, n):  
        if sum(diviseurs_propres(i)) == i:  
            L.append(i)  
    return L
```