Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

PI2T Développement informatique

Examen Juin 2016

Solutionnaire

Les réponses aux exercices dont la réponse est un code à fournir sont données à titre d'exemple, plusieurs solutions pouvant en effet répondre aux problèmes posés.

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

1 Vrai ou faux (3 points)

(Connaissances générales)

Pour chacune des affirmations suivantes, indiquez si elle est vraie ou fausse. *Si vous avez répondu correctement à toutes les affirmations d'un bloc, vous obtenez un point, si vous avez fait une faute vous obtenez un demi-point et sinon vous avez zéro (une abstention compte comme une faute).*

1.1 Gestion de projet et réseau

Affirmation	**Vrai**	**Faux**
A/ Avec Git, plusieurs copies intégrales d'un projet informatique peuvent exister en même temps sur plusieurs machines.	☑	☐
B/ Un test fonctionnel a pour but de vérifier que l'utilisateur d'un programme parvienne à retrouver facilement et rapidement les différentes fonctions du programme.	☐	☑
C/ Un service comme BitTorrent se base sur une architecture client-serveur.	☐	☑
D/ Deux applications connectées qui communiquent sur un réseau sont chacune identifiée par une adresse IP et un numéro de port.	☑	☐
E/ Lorsqu'on utilise un système de gestion de versions, on est limité à un seul commit par jour.	☐	☑

1.2 Algorithmique

Affirmation	**Vrai**	**Faux**
A/ Pour un algorithme donné, il n'existe qu'un et un seul programme Python correspondant.	☐	☑
B/ Toute fonction récursive finit toujours par s'arrêter.	☐	☑
C/ Tous les nœuds d'un arbre possèdent un parent.	☐	☑
D/ La postcondition d'une fonction représente les conditions qui seront toujours satisfaites après exécution de la fonction.	☐	☑
E/ Une pile est une file.	☐	☑

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

1.3 Programmation concurrente et fonctionnelle

Affirmation	**Vrai**	**Faux**
A/ Concurrence et parallélisme sont deux synonymes.	☐	☑

B/ L'exécution du programme suivant affiche 4.
`x = lambda x: x + 2`
`x = x(2)`
`print(x)`
	☑	☐

C/ La fonction prédéfinie sum permet de concaténer tous les éléments d'une liste de chaines de caractères.	☐	☑
D/ Il est possible d'exécuter des programmes de manière concurrente sur une machine qui ne peut exécuter physiquement qu'une seule instruction à la fois.	☑	☐
E/ Une fonction Python est un objet que l'on peut stocker dans une variable.	☑	☐

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

2 Théorie (4 points)

(Compréhension d'un concept informatique)

Expliquez le concept de **système de gestion de versions** et son intérêt dans le cadre du développement d'un programme informatique, seul et à plusieurs.

*Un **système de gestion de versions** est un logiciel qui permet de sauvegarder l'**historique complet des modifications** apportées à un code source. Les modifications sont organisées sous forme de **commit**, objet reprenant un ensemble de modifications apportées sur un ensemble de fichiers. On peut à tout moment naviguer parmi ces différents commit, retraçant ainsi l'histoire du code source d'un programme et de ses différentes versions. En outre, l'utilisation d'un tel logiciel permet également d'avoir une **stratégie de backup** de son code.*

*Dans le cadre du développement d'un programme informatique, un tel outil permet de structurer le travail de développement et d'organiser le travail à plusieurs développeurs. Il est en effet possible, avec un tel système, d'avoir **plusieurs copies complètes du projet** situées sur différentes machines. Chaque développeur peut ainsi travailler de son côté, avant de synchroniser l'ensemble de ses modifications avec les autres développeurs (il lui suffit d'envoyer les commits qu'il a réalisé et de récupérer ceux des autres). Un système de gestion de versions supporte également l'utilisation de **branches** de développement, permettant au programme informatique d'avoir plusieurs versions différentes développées en parallèle (pour différents systèmes d'exploitation par exemple).*

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

3 Expression régulière (3 points)

(Niveau élémentaire)

Définissez une expression régulière permettant de vérifier les motifs suivants :

1. Une chaine de caractères donnée doit être un code postal constitué de quatre ou cinq chiffres arabes, et tel que le premier (celui tout à gauche) est non nul.
2. Recherche de mots répétés à l'intérieur d'une chaine de caractères donnée (deux mots exactement les mêmes qui se suivent).
3. Une chaine de caractères donnée doit représenter un prix, c'est-à-dire un nombre naturel, suivi éventuellement d'un point suivi d'une séquence de chiffres (de 0 à 9).

1. Le premier chiffre doit être non nul, puis on lui adjoint entre 3 et 4 chiffres.

```
[1-9][0-9]{3,4}
```

2. On cherche d'abord un mot qu'on va capturer, puis on cherche le mot capturé après un caractère blanc.

```
(\w+)\s\1
```

3. Il faut ici distinguer deux cas, et donc écrire deux expressions régulières, selon que la partie entière est nulle ou non. Pour le premier cas, où la partie entière est non nulle, cette dernière ne peut pas commencer par zéro.

```
(?:[1-9][0-9]*(?:\.[0-9]+)?)|(?:0\.[0-9]+)
```

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

4 Trouver l'erreur (5 points)

(Niveau moyen)

(A) Lorsqu'on exécute le programme suivant, l'interpréteur Python indique un problème au niveau de l'expression régulière.

```
1 import re
2
3 pattern = '[A-Z]:(?:\\[^<>:"/\|?*]+)*\\?'
4 p = re.compile(pattern)
5
6 print(p.match("C:\\Users\\lur\\Desktop"))  # devrait matcher
7 print(p.match("/home/lur/Desktop"))        # ne devrait pas matcher
```

Cette expression régulière doit matcher les *« chemins Windows »*. Sachant que les noms de fichiers Windows ne peuvent pas contenir les caractères `<>:"/\|?*`, identifiez le problème et corrigez-le. Justifiez votre réponse.

Le souci qui se produit ici concerne la définition de l'expression régulière à l'aide d'une chaine de caractères Python. En effet, le caractère `\` *qu'on veut utiliser pour matcher les chemins Windows est également utiliser pour définir des séquences d'échappement, pour les chaines de caractères Python (par exemple* `\n` *pour un retour à la ligne), mais également pour les expressions régulières (par exemple* `\.` *pour capturer le caractère « point »).*

La solution consiste à utiliser une chaine de caractères brute :

```
pattern = r'[A-Z]:(?:\\[^<>:"/\|?*]+)*\\?'
```

Pourquoi faut-il malgré tout encore doubler les backslashs ? Car si on avait par exemple mis `\?`*, cela revient à capturer un point d'interrogation et pas un ou zéro backslash. Une autre solution possible, sans utiliser les chaines de caractères brutes, est :*

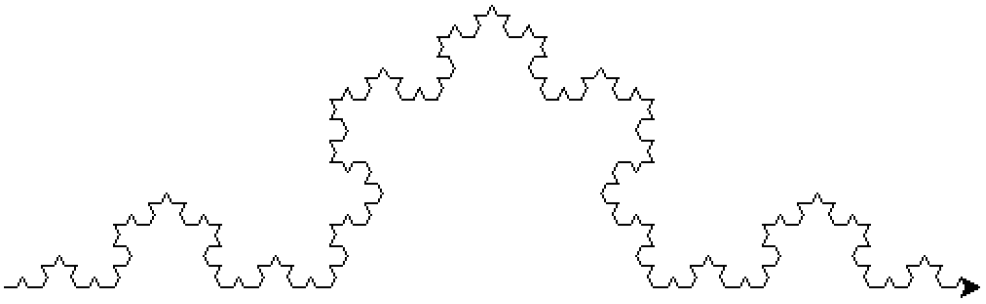
```
pattern = '[A-Z]:(?:\\\\[^<>:"/\|?*]+)*\\\\?'
```

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

(B) La fonction récursive suivante devrait dessiner la fractale représentée ci-dessous, mais l'interpréteur Python indique un problème de récursion infinie. Expliquez à quoi est due cette erreur et comment on pourrait la corriger.



```
1  from turtle import *
2
3  def vonkoch(dist):
4      if dist == 5:
5          forward(dist)
6      else:
7          vonkoch(dist/3)
8          left(60)
9          vonkoch(dist/3)
10         right(120)
11         vonkoch(dist/3)
12         left(60)
13         vonkoch(dist/3)
14
15 vonkoch(200)
16 done()
```

Une fonction récursive se termine lorsqu'on ne fait plus d'appel récursif, c'est-à-dire lorsqu'on atteint le ***cas de base****. Dans la fonction* vonkock*, le cas de base correspond à la condition* dist == 5*, c'est-à-dire que le paramètre de la fonction vaut exactement 5. Les appels récursifs se font en divisant la valeur de* dist *par trois, sachant qu'on commence avec une valeur initiale de* 200*. Les appels successifs seront donc :* vonkoch(200)*,* vonkoch(66.66...)*,* vonkoch(22.22...)*,* vonkoch(7.407407...)*,* vonkoch(2.469...)

On atteindra donc jamais exactement la valeur 3, et la récursion ne finira donc jamais. Une solution immédiate pour résoudre le problème consiste à remplacer le cas de base par :

```
if dist <= 5:
```

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

5 Multiplier des matrices (5 points) (Niveau avancé)

Le programme suivant sert a multiplier des matrices. Il le fait en utilisant la fonction map qui, comme vous le savez, prend une fonction en paramètre. Dans ce programme, c'est la fonction getRowMultiplier qui génère la fonction passée en paramètre à map.

```
1  # In this program, a matrix is a list of rows and a row
2  # is a list of values
3
4  # As a matrix is a list of row
5  # We use the transpose operation to get a list of columns.
6  # That will be useful in the mult function
7  #
8  # - params: a matrix as described before
9  # - return: the transposed matrix (aka. the list of columns)
10 def transpose(matrix):
11     trans = []
12     for j in range(len(matrix[0])):
13         trans.append([])
14         for i in range(len(matrix)):
15             trans[j].append(matrix[i][j])
16     return trans
17
18 # This function return a function that multiply by a single row
19 # It is used with a map function for matrix multiplication
20 #
21 # - params: a row
22 # - return: a function that take a column (list of values) and
23 #           return the result of (row * col)
24 def getRowMultiplier(row):
25     # À compléter...
26
27 # In a matrix multiplication each row of the first matrix
28 # multiply each column of the second one. We can multiply a row
29 # by all the columns at once by using a map function. Each call
30 # to map need a function that multiply by a specific row.
31 # The getRowMultiplier function generate these functions.
32 #
33 # - params: two matrix, the width of the first must match the
34 #           height of the second
35 # - return: a matrix that is the product of the two params
36 # - raise: ArithmeticError if matrix sizes are incompatible
37 def mult(A, B):
38     if len(A[0]) != len(B):
39         raise ArithmeticError("Incompatible Matrix sizes")
40     res = []
41     columnsOfB = transpose(B) # get the list of columns
42     for row in A:
43         res.append(list(map(getRowMultiplier(row), columnsOfB)))
44     return res
45
46
47 matrix = [
48     [1,2],
49     [3,4]
50 ]
51 print(mult(matrix, matrix))
```

Vous devez écrire la fonction getRowMultiplier.

Nom : 07/06/2016 13:30

Matricule : (120 minutes)

PI2T — Développement informatique

On doit définir une nouvelle fonction à qui on passe en paramètre une colonne d'une matrice et qui multiplie cette colonne par la ligne `row` *reçue en paramètre par la fonction* `getRowMultiplier`*. Cette nouvelle fonction calcule en fait simplement le produit scalaire entre* `row` *et* `col`*, que l'on fait avec une boucle.*

```
def getRowMultiplier(row):
    def multiplier(col):
        res = 0
        for i in range(len(row)):
            res += row[i] * col[i]
        return res
    return multiplier
```