Nom : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

PI2T Développement informatique

Examen Aout 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 : 19/08/2016 08:30

Matricule : (105 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/ La commande `git commit` crée une nouvelle entrée dans l'historique du projet, reprenant tous les changements qui ont été placés dans la zone de transit.	☑	☐
B/ Travis est une plateforme en ligne de déploiement automatique de code.	☐	☑
C/ Dans une architecture peer-to-peer, une des machines du réseau est placée au centre et sert de relais pour toutes les communications entre deux machines du réseau.	☐	☑
D/ Afin de communiquer à l'aide du protocole TCP, une fois le socket créé, il faut le faire écouter sur un port donné à l'aide de la méthode `bind`	☐	☑
E/ Deux applications connectées qui communiquent sur un réseau sont chacune identifiée par une adresse IP et un numéro de port.	☑	☐

1.2 Algorithmique

Affirmation	**Vrai**	**Faux**
A/ La postcondition d'une fonction représente les conditions qui seront toujours satisfaites après exécution de la fonction, pour autant que les préconditions étaient satisfaites.	☑	☐
B/ Il faut toujours un cas de base dans une fonction récursive.	☑	☐
C/ Tous les nœuds d'un arbre possèdent au moins un enfant.	☐	☑
D/ Pour un algorithme donné, il n'existe qu'un et un seul programme Python correspondant.	☐	☑
E/ Une liste Python est une file.	☑	☐

Nom : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

1.3 Programmation concurrente et fonctionnelle

Affirmation	**Vrai**	**Faux**
A/ La fonction prédéfinie sum permet de calculer la somme des éléments d'une liste de nombres entiers.	☑	☐
B/ Une fonction Python est un objet que l'on peut stocker dans une variable.	☑	☐

C/ L'exécution du programme suivant pourrait afficher <function <lambda> at 0x107b81048>.
`x = lambda x: x + 2`
`x = x(2)`
`print(x)`
	☐	☑

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/ Concurrence et parallélisme sont deux synonymes.	☐	☑

Nom : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

2 Théorie (4 points)

(Compréhension d'un concept informatique)

Présentez les **architectures peer-to-peer et client/serveur** en les comparant et en mettant en évidence leurs avantages et inconvénients.

Il y a principalement deux types d'architecture pour relier ensemble plusieurs machines dans un réseau :

- *Dans une architecture peer-to-peer, le même programme est exécuté sur plusieurs machines qui peuvent être connectées avec n'importe quelle autre machine du réseau. On a, par exemple, des systèmes comme Bittorrent, Spotify ou Bitcoin.*
- *Dans une architecture client/serveur, certaines machines sont des clients et d'autres sont des serveurs. Les clients doivent d'abord se connecter aux serveurs, leur envoyer une requête et ces derniers y répondent ensuite. On a, par exemple, des systèmes comme HTTP ou FTP.*

Un avantage d'un réseau peer-to-peer réside dans sa simplicité. Toutes les machines exécutent en effet le même programme et peut se connecter à n'importe laquelle autre. Si un machine du réseau tombe, le reste du réseau peut continuer à fonctionner.
Un avantage d'un réseau client/serveur est qu'une fois la connexion établie entre un client et un serveur, on a une ligne de transmission fiable et ces derniers peuvent communiquer entre eux tant que cette connexion reste ouverte. Typiquement, seuls les serveurs doivent être des machines puissantes qui peuvent ainsi servir de nombreux clients.
Un désavantage d'un réseau peer-to-peer est que si le programme à exécuter est lourd, il faudra que toutes les machines du réseau soient puissantes. La mise à jour du programme est également plus complexe car elle doit se faire partout.
Un désavantage d'un réseau client/serveur est que si le serveur tombe, les clients ne peuvent plus rien faire. Il faut donc typiquement prévoir plusieurs serveurs.

Nom : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

3 Expression régulière (3 points)

(Niveau élémentaire)

Définissez une expression régulière permettant de vérifier les motifs suivants :

1. Un numéro de matricule constitué de cinq chiffres arabes, et tel que le premier (celui tout à gauche) est non nul, *par exemple 92837.*
2. Une date avec le jour et le mois sur deux chiffres arabes, puis l'année sur quatre chiffres arabes, le tout séparé avec des /, *par exemple 18/01/2016 ou 02/02/1992.*
3. Un prix, c'est-à-dire un nombre naturel, suivi éventuellement d'un point suivi d'une séquence de chiffres (de 0 à 9), *par exemple 0.15 ou 170.*

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

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

2. Le premier chiffre du jour est 0, 1, 2 ou 3 et le second peut être n'importe quel chiffre. Le premier chiffre du mois est 0 ou 1 et le second peut-être n'importe quoi. Pour l'année, on considère quatre chiffres, tel que le premier n'est pas nul.

```
[0123][0-9]/[01][0-9]/[1-9][0-9]{3}
```

On suppose qu'on vérifiera ensuite que le jour doit être inférieur à 31 *et que le mois est bien compris entre* 1 *et* 12*, et que l'année est valide.*

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 : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

4 Trouver l'erreur (5 points)

(Niveau moyen)

(A) La fonction line ci-dessous permet de représenter une droite dans le plan d'équation $ax + b$. Ce que l'on souhaite faire, c'est afficher la valeur en $x = 12$ de la droite $f(x) = 2x + 3$:

```
1 def line(a, b):
2     def f(x):
3         return a * x + b
4     return f
5
6 print(f(2, 3, 12))
```

Le programme ci-dessus provoque une erreur à la ligne 6, lors de l'appel. Identifiez le problème et corrigez-le (vous devez remplacer la ligne 6 par une ou plusieurs instructions). Justifiez votre réponse.

La fonction f est définie en interne dans la fonction line et donc pas accessible pour l'appel fait en ligne 6. Ce qu'il faut faire, c'est appeler la fonction line, avec les bons paramètres, et récupérer la fonction qu'elle renvoie comme valeur de retour. On peut ensuite évaluer la fonction au point 12, à l'aide de cette valeur récupérée :

```
f = line(2, 3)
print(f(12))
```

On pourrait aussi tout faire en une fois :

```
print(line(2, 3)(12))
```

(B) La fonction récursive suivante devrait compter combien de fois on peut diviser un nombre entier donné par deux avant d'atteindre zéro, sachant qu'on considère des divisions entières. Le problème est qu'elle ne calcule pas toujours le bon résultat. Expliquez pourquoi cette fonction est fausse et comment on pourrait la corriger.

```
1 def count(n):
2     if n != 0:
3         return 0
4     if n % 2 == 0:
5         return 1 + count(n / 2)
6     return 1 + count((n - 1) / 2)
```

On a un problème avec le premier cas de base qui sera toujours satisfait. Du coup, pour toute valeur positive avec laquelle on appelle cette fonction, on reçoit 0 comme valeur de retour. La condition est en fait opposée et, pour corriger le code, il suffit donc de remplacer la ligne 2 par :

```
if n == 0:
```

Nom : 19/08/2016 08:30

Matricule : (105 minutes)

PI2T — Développement informatique

5 Filtrage de listes (5 points) (Niveau avancé)

Soit le programme suivant qui définit une fonction filter qui permet de filtrer les données d'une liste. Cette fonction reçoit une liste et une fonction booléenne qui décide s'il faut garder ou non l'élément.

```
1  def even(x):
2      # À compléter...
3
4  def lower(a):
5      # À compléter...
6
7  def filter(data, condition):
8      result = []
9      for elem in data:
10         if condition(elem):
11             result.append(elem)
12     return result
13
14 data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
15 print(filter(data, even))                # doit afficher [2, 4, 6, 8, 10]
16 print(filter(data, lower(3)))            # doit afficher [1, 2]
17 print(filter(data, lower(5)))            # doit afficher [1, 2, 3, 4]
```

Vous devez écrire les fonctions even et lower. La première fonction est directement le filtre et doit tester si le x reçu en paramètre est pair ou non. La seconde fonction reçoit un paramètre a et doit renvoyer une fonction qui représente un filtre qui teste si le x qu'elle reçoit est strictement plus petit que a.

La fonction even renvoie donc un booléen qui teste si la valeur du x reçu en paramètre est pair ou non :

```
def even(x):
    return x % 2 == 0
```

La fonction lower renvoie une fonction qui teste si la valeur du x reçu en paramètre est strictement plus petit que a ou non :

```
def lower(a):
    def f(x):
        return x < a
    return f
```