

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique****PI2T Développement informatique***Examen Aout 2016***Consignes générales**

- Pas de calculatrice.
- Feuilles de brouillon incluses, à rendre avec sa copie.
- Réponses au bic ou stylo, noir ou bleu.

**Explications de vos réponses**

- Vous pouvez ajouter des commentaires pour expliquer des parties non triviales de vos réponses.
- N'hésitez pas à mentionner toute hypothèse non mentionnée dans l'énoncé que vous auriez prise.
- Parfois un petit schéma vaut mieux qu'un long discours.

**Réservé aux correcteurs (questionnaire A)**

<b>Q1</b>						/3
<b>Q2</b>						/4
<b>Q3</b>						/3
<b>Q4</b>						/5
<b>Q5</b>						/5
<b>Total</b>						<b>/20</b>

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique**

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 <code>git commit</code> crée une nouvelle entrée dans l'historique du projet, reprenant tous les changements qui ont été placés dans la zone de transit.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Travis est une plateforme en ligne de déploiement automatique de code.	<input type="checkbox"/>	<input type="checkbox"/>
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.	<input type="checkbox"/>	<input type="checkbox"/>
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 <code>bind</code>	<input type="checkbox"/>	<input type="checkbox"/>
E/ Deux applications connectées qui communiquent sur un réseau sont chacune identifiée par une adresse IP et un numéro de port.	<input type="checkbox"/>	<input type="checkbox"/>

**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.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Il faut toujours un cas de base dans une fonction récursive.	<input type="checkbox"/>	<input type="checkbox"/>
C/ Tous les nœuds d'un arbre possèdent au moins un enfant.	<input type="checkbox"/>	<input type="checkbox"/>
D/ Pour un algorithme donné, il n'existe qu'un et un seul programme Python correspondant.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Une liste Python est une file.	<input type="checkbox"/>	<input type="checkbox"/>

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 <code>sum</code> permet de calculer la somme des éléments d'une liste de nombres entiers.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Une fonction Python est un objet que l'on peut stocker dans une variable.	<input type="checkbox"/>	<input type="checkbox"/>
C/ L'exécution du programme suivant pourrait afficher <code>&lt;function &lt;lambda&gt; at 0x107b81048&gt;</code> . <pre>x = lambda x: x + 2 x = x(2) print(x)</pre>	<input type="checkbox"/>	<input type="checkbox"/>
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.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Concurrence et parallélisme sont deux synonymes.	<input type="checkbox"/>	<input type="checkbox"/>

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.

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique***(suite de la réponse)*

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*.

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.



Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique**

(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)
```

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`.

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique***(suite de la réponse)*

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique**

**Brouillon**

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique**

*(brouillon suite)*

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

**PI2T — Développement informatique***(brouillon suite)*

Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

## PI2T — Développement informatique

## Commandes Bash de base

## Général

`clear` efface le contenu de la fenêtre  
`history` affiche l'historique des dernières commandes  
`exit` quitte le terminal

## Information système

`whoami` affiche l'identité de l'utilisateur courant  
`uptime` information sur la durée de fonctionnement  
`date` affiche la date courante  
`cal` affiche le calendrier du mois courant  
`man <cmd>` affiche l'aide de la commande `cmd`  
*(quitter l'aide avec la touche q)*

information sur le système d'exploitation  
 liste les utilisateurs connectés  
 affiche l'utilisation des disques  
 tailles lisibles par un humain

## Navigation

`pwd` affiche le dossier courant  
`ls` liste le contenu du dossier courant  
`ls <path>` liste le contenu du dossier `path`  
`ls -a` affiche tous les fichiers et dossiers  
`ls -lh` liste détaillée et tailles lisibles par un humain  
`cd <path>` change le dossier courant en `path`  
`cd ..` remonte dans le dossier parent

## Manipulation de fichiers

`cat <path>` affiche le contenu du fichier `path`  
`head <path>` affiche les 10 premières lignes du fichier `path`  
`head -n <nb> <path>` affiche les `nb` premières lignes du fichier `path`  
`tail <path>` affiche les 10 dernières lignes du fichier `path`  
`tail -n <nb> <path>` affiche les `nb` dernières lignes du fichier `path`

`mkdir <path>` crée un nouveau dossier `path`  
`touch <path>` crée un nouveau fichier vide `path`  
`cp <path1> <path2>` copie le fichier `path1` vers le fichier `path2`  
`cp <path1> <path2> /` copie le fichier `path1` dans le dossier `path2`  
`cp -R <path1> <path2>` copie le dossier `path1` vers le dossier `path2`  
`mv <path1> <path2> /` déplace le fichier `path1` dans le dossier `path2`  
`mv <path1> <path2>` déplace le dossier `path1` vers `path2`  
`rm <path>` supprime le fichier `path`  
`rm -i <path>` demande confirmation avant suppression  
`rm -r <path>` supprime le dossier `path`  
`find <path> -name <name>` cherche le fichier `name` dans le dossier `path`

## Gestion des processus

`ps` liste statique des processus en cours d'exécution de l'utilisateur courant  
`ps -a` liste les processus de tous les utilisateurs  
`ps -e` liste tous les processus  
`ps -f` affiche toutes les informations des processus  
`top` liste dynamique des processus en cours d'exécution (*quitter la liste avec la touche q*)  
`kill <pid>` tue le processus numéro `pid`  
`kill -9 <pid>` tue violemment le processus

## Divers

`diff <path1> <path2>` affiche la différence entre les deux fichiers  
`wc <path>` affiche le nombre de lignes, de mots et de caractères du fichier `path`  
`lpr <path>` imprime le fichier `path`  
`lpq` affiche les files d'impression  
`wget <url>` télécharge le fichier `url`  
`ping <url>` envoie une requête ping au serveur `url`  
`ifconfig` montre la configuration des interfaces réseau



Nom :

19/08/2016 08:30

Matricule :

(105 minutes)

## PI2T — Développement informatique

## Syntaxe des expressions régulières en Python

Il y a 14 *méta-caractères* à précéder d'un backslash pour les capturer :

. ~ \$ \* + ? { [ ] \ | ( )

## Classe de caractères

Représente un ensemble de caractères :

[abc] a, b et c  
[~abc] tout sauf a, b et c  
[0-9] tous les caractères compris entre 0 et 9  
[a-zA-Z] toutes les lettres de l'alphabet romain

Il existe des classes prédéfinies :

. n'importe quel caractère (sauf retour à la ligne)  
\d un chiffre décimal ([0-9])  
\s un caractère blanc ([ \t\n\r\f\v])  
\w un caractère alphanumérique ([a-zA-Z0-9\_])  
complémentaires des ensembles précédents  
\D, \S et \W

## Répétition

Permet de matcher plusieurs fois une même expression régulière  $R$  :

$R\{n\}$  exactement  $n$  occurrences  
 $R\{m, n\}$  au moins  $m$  et au plus  $n$  occurrences  
 $R\{m, \}$  au moins  $m$  occurrences  
 $R\{, n\}$  au plus  $n$  occurrences

Il existe des répétitions prédéfinies :

$R^*$  zéro ou plusieurs occurrences ( $\{0, \}$ )  
 $R^+$  une ou plusieurs occurrences ( $\{1, \}$ )  
 $R?$  zéro ou une occurrence ( $\{0, 1\}$ )

Par défaut, la chaîne analysée sera consommée au maximum possible avec les répétitions. Pour ne pas être glouton et consommer le moins possible, on ajoute un  $?$  en suffixe :

$\{m, n\}?$   $^*$   $^+$   $?$

## Frontière de recherche

^ début de la ligne  
\$ fin de la ligne  
\A début de la chaîne de caractères  
\Z fin de la chaîne de caractères  
\b limite d'un mot (début ou fin)  
\B pas la limite d'un mot

Les frontières ne consomment aucun caractères dans la chaîne, elles représentent simplement une assertion.

## Alternative

$R_1 | R_2$  choix entre les deux expressions régulières  $R_1$  et  $R_2$

## Groupe

Permet de délimiter une sous-expression régulière  $R$ , pour par exemple la répéter ou pour former une alternative :

$(R)$  groupe capturant  
 $(?:R)$  groupe non-capturant  
 $(?P=nameR)$  groupe capturant nommé  $name$

Il est possible de faire référence à la valeur capturée par un groupe précédent :

$\backslash i$  valeur capturée par le groupe numéro  $i$   
 $(?P=name)$  valeur capturée par le groupe nommé  $name$

## Lookahead

Permet de tester une assertion  $R$ , sans avancer dans la chaîne :

$(?=R)$  succès si  $R$  matche à la position courante  
 $(?!R)$  succès si  $R$  ne matche pas à la position courante

