

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**IC1T Programmation***Examen Janvier 2016***Consignes générales**

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

Mise en forme du code

- Vous pouvez ajouter des commentaires pour expliquer des parties non triviales de vos réponses.
- Faites attention au soin de vos réponses, et en particulier à rendre l'indentation explicite.
- Ne pas recopier les entêtes des fonctions déjà fournies.

Réservé aux correcteurs (questionnaire A)

Q1								/3
Q2								/4
Q3								/7
Q4								/6
Total								/20

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication

Aide-mémoire

Fonctions prédéfinies

Fonction	Description
<code>print()</code>	Affiche (on peut utiliser les paramètres <code>end</code> et <code>sep</code>)
<code>sqrt()</code>	Racine carrée

Pour les fonctions `math` : `from math import *`

Conditions

```
a = -5
if a == 0:
    print("a est nul")
elif a < 0:
    print("a est négatif")
else:
    print("a est positif")
```

Opération	Description
<code>< et ></code>	strictement inférieur ou supérieur
<code><= et >=</code>	«inférieur ou égal» ou «supérieur ou égal»
<code>== et !=</code>	égal ou différent
<code>and et or</code>	«et» logique ou «ou» logique
<code>not</code>	« non » logique
<code>in</code>	vérifie qu'une valeur appartient à une séquence

Boucles

```
i = 0
while i < 5:
    i = i + 1
```

```
for variable in séquence:
    bloc d'instructions
```

Fonctions

```
# création de la fonction
def nomDeLaFonction(paramètres):
    bloc d'instructions

# appel de la fonction
nomDeLaFonction(valeurs des paramètres)
```

Pour définir une valeur de retour dans le bloc d'instructions, on utilise le mot-clé `return`

Turtle

Pour importer le module : `from turtle import *`

Fonctions	Descriptions
<code>forward(distance)</code>	Avancer d'une distance donnée
<code>backward(distance)</code>	Reculer d'une distance donnée
<code>up()</code>	Relever le crayon
<code>down()</code>	Abaisser le crayon
<code>color(couleur)</code>	Changer la couleur du crayon. couleur en anglais
<code>left(angle)</code>	Tourner à gauche d'un angle donné (en degrés).
<code>right(angle)</code>	Tourner à droite d'un angle donné (en degrés).
<code>width(épaisseur)</code>	Changer l'épaisseur du crayon.
<code>goto(x, y)</code>	Aller à l'endroit de coordonnées x, y
<code>reset()</code>	Effacer tout
<code>done()</code>	Garder la fenêtre ouverte.

Séquences

```
jours = ['lundi', 'mardi', 'vendredi']

print(jours[1]) # affichera mardi
jours[0] = 'monday' # modifie le premier élément

len(jours) # renvoie la longueur de la liste
del(jours[1]) # supprime l'élément d'indice 1

jours[0:2] # renvoie une liste avec les éléments de 0
            compris à 2 non-compris
jours[2:2] = ['mercredi', 'jeudi'] # insère deux éléments
jours[:0] = ['avant'] # insère en début de liste
jours[len(L):] = ['après'] #insère en fin de liste
```

Opérations	Descriptions
<code>+</code>	Concatène deux séquences
<code>«entier» *</code>	répète une séquence

Les chaînes de caractères sont des séquences non-modifiables.

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**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 Bases du Python

Affirmation	Vrai	Faux
A/ Le programme suivant affiche Hello à l'écran lors de son exécution. <pre>Hello = 42 print(Hello)</pre>	<input type="checkbox"/>	<input type="checkbox"/>
B/ On peut changer le nom d'une variable déjà existante.	<input type="checkbox"/>	<input type="checkbox"/>
C/ Le programme suivant provoque une erreur lors de son exécution. <pre>x = 12 x = 'Hello'</pre>	<input type="checkbox"/>	<input type="checkbox"/>
D/ Le nombre 12.24 est une donnée de type int.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Soient les variables x et y stockant des nombres entiers valant respectivement 12 et -7, l'expression suivante vaut False. <pre>x > -7 and not y >= 0</pre>	<input type="checkbox"/>	<input type="checkbox"/>

1.2 Instruction conditionnelle et itérative

Affirmation	Vrai	Faux
A/ Une boucle while finit toujours pas s'arrêter.	<input type="checkbox"/>	<input type="checkbox"/>
B/ Une condition est une expression de type booléen.	<input type="checkbox"/>	<input type="checkbox"/>
C/ Le programme suivant affiche Faux lors de son exécution. <pre>x = 12 if -x > 0: print("Vrai") else: print("Faux")</pre>	<input type="checkbox"/>	<input type="checkbox"/>
D/ On est toujours obligé de déclarer un bloc else lorsqu'on a un if.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Le programme suivant affiche 42 lors de son exécution. <pre>x = 42 if x = 42: print(42)</pre>	<input type="checkbox"/>	<input type="checkbox"/>

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**1.3 Séquence**

Affirmation	Vrai	Faux
A/ Le programme suivant affiche tic-tac à l'écran lors de son exécution. <pre>word = "tic-tic" word[5] = "a" print(word)</pre>	<input type="checkbox"/>	<input type="checkbox"/>
B/ Soit une liste non vide stockée dans une variable data . On peut afficher la longueur de la liste à l'écran avec l'instruction suivante. <pre>print(data.len())</pre>	<input type="checkbox"/>	<input type="checkbox"/>
C/ Soit une liste non vide stockée dans une variable data . On peut afficher le premier élément de la liste avec l'instruction suivante. <pre>print(data[1])</pre>	<input type="checkbox"/>	<input type="checkbox"/>
D/ Soit une liste non vide stockée dans une variable data . L'expression data[: -1] permet d'extraire le dernier élément de la liste.	<input type="checkbox"/>	<input type="checkbox"/>
E/ Le programme suivant affiche 15 lors de son exécution. <pre>sum = 0 for i in range(5): sum += i print(sum)</pre>	<input type="checkbox"/>	<input type="checkbox"/>

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**2 Nombre de mots (4 points)**

(Niveau élémentaire)

Définir une fonction qui renvoie le nombre de mots qu'il y a dans une chaîne de caractères qu'elle reçoit en paramètre. **Les mots sont séparés par un seul espace.** De plus, on suppose que la chaîne de caractères ne contient que des lettres et des espaces, et donc aucune ponctuation, et aucun espace en début ou fin de chaîne.

```
def countwords(sentence) :
```

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**3 Fusion de listes (7 points)**

(Niveau moyen)

Définir une fonction qui prend deux listes non vides d'entiers en paramètres et qui renvoie une liste contenant tous les éléments des deux listes de départ en alternance. Si une liste est plus longue que l'autre, ses éléments sont ajoutés à la fin de la liste à renvoyer.

Par exemple, si $L1 = [1, 2, 3, 4]$ et $L2 = [0, 9, 8, 7, 6, 5]$,

alors la fonction doit renvoyer la liste $[1, 0, 2, 9, 3, 8, 4, 7, 6, 5]$.

def merge(L1, L2) :

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**4 Exponentiation (6 points)**

(Niveau avancé)

Définir une fonction qui prend deux nombres entiers positifs, une base x et un exposant n , en paramètres et renvoie la valeur de x^n en se basant sur les propriétés récursives suivantes :

$$\begin{cases} x^0 = 1 \\ x^n = x^{n/2} \cdot x^{n/2} & (\text{si } n \text{ est pair}) \\ x^n = x^{n-1} \cdot x & (\text{si } n \text{ est impair}) \end{cases}$$

Veillez à ne pas calculer plusieurs fois les mêmes valeurs (en particulier ne pas calculer deux fois $x^{n/2}$).

def power(x, n) :

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication**Brouillon**

Nom :

12/01/2016 08:30

Matricule :

(135 minutes)

B1030 — Informatique et communication*(brouillon suite)*