

Examen du module d'ingénierie informatique

Durée : 1 heure 30

Documents autorisés : 1 feuille A4 recto-verso

Exercice 1 (2 points) — Question de cours

Quel est l'affichage obtenu après l'exécution du code Python ci-dessous. Ne justifiez pas votre réponse.

```
def fonction(x) :  
    x = x + 2  
  
x = 1  
fonction(x)  
print("x =", x)
```

Exercice 2 (3 points) — Affichages

Écrivez une fonction Python `affichage(painter, tab, n, image)` prenant en argument un `Painter` `PyQt5`, un tableau numpy 1D `tab` de taille `n`, où `n` est un entier **pair strictement positif**, ainsi qu'un `QPixmap` `image`. Les éléments de `tab` représentent des coordonnées en pixels. La fonction `affichage` a pour but d'afficher l'image aux coordonnées spécifiées dans `tab` (ce sont celles des coins supérieurs gauches des images). Par exemple, si `tab` correspond au tableau

3	5	10	7
---	---	----	---

, la fonction affiche une première fois l'image dans la fenêtre aux coordonnées (3,5) et une deuxième fois aux coordonnées (10,7).

Exercice 3 (3 points) — Test FizzBuzz

Écrivez une fonction Python `fizzBuzz()` qui affiche tous les nombres entiers n de 1 à 100 (1 nombre par ligne), excepté dans les cas suivants :

- lorsque n est divisible par 3, la fonction doit afficher `Fizz`,
- lorsque n est divisible par 5, la fonction doit afficher `Buzz`,
- lorsque n est divisible par 3 et 5, la fonction doit afficher `FizzBuzz`.

Ainsi, les affichages de votre fonction débutent par la séquence ci-dessous où, pour des raisons évidentes de place dans l'énoncé, les retours à la ligne ont été remplacés par des espaces :

1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 ...

Exercice 4 (3 points) — Consécutivement votre

Écrivez une fonction Python `nb_consecutifs(tab, n, x)` qui prend en argument un tableau numpy 1D `tab` de taille $n > 0$ ainsi qu'un indice `x` de ce tableau. La fonction renvoie le nombre d'éléments consécutifs voisins de `x` qui ont exactement la même valeur que l'élément d'indice `x`. Par exemple, si

`tab` est égal au tableau

1	3	6	7	3	3	3	3	2	5
---	---	---	---	---	---	---	---	---	---

 et si `x = 5`, alors la fonction doit renvoyer 4 car `tab[5] = 3` et il y a précisément 4 éléments consécutifs de valeur 3 « collés » à l'élément d'indice 5 (ici, même si le deuxième élément du tableau a pour valeur 3, il n'est pas comptabilisé puisqu'il est séparé des autres 3 par les chiffres 6 et 7).

Exercice 5 (4 points) — *MinCols*

Écrivez une fonction Python `minCol(tab, nbLigs, nbCols)` qui prend en argument un tableau numpy 2D `tab` de taille `nbLigs` lignes et `nbCols` colonnes. Les lignes sont la 1ère dimension du tableau, comme dans CandyCrush, et on suppose que `nbLigs > 0` et `nbCols > 0`. La fonction doit renvoyer un nouveau tableau numpy 1D dont les éléments correspondent aux nombres les plus petits de chaque colonne. Par exemple, si l'on a `tab =`

2	5	12
4	-3	0

, alors la fonction renverra le tableau

2	-3	0
---	----	---

Exercice 6 (2 points) — *Descente de ligne 1*

Écrivez une fonction Python `descente(tab, nbLigs, nbCols)` qui prend en argument un tableau numpy 2D `tab` de taille `nbLigs` lignes et `nbCols` colonnes. Les lignes sont la 1ère dimension du tableau et on suppose que `nbLigs > 0` et `nbCols > 0`. La fonction modifie le tableau en faisant descendre d'une ligne chaque ligne du tableau et elle place une ligne de 0 sur la plus haute. Par exemple, après exécution de la fonction, le tableau

1	3	5	6	1	deviendra	0	0	0	0	0
2	5	4	9	4		1	3	5	6	1
3	8	1	0	7		2	5	4	9	4
4	9	3	7	7		3	8	1	0	7

La fonction `descente` ne renvoie rien, elle se contente de modifier le tableau `tab`.

Exercice 7 (3 points) — *Retournement de situation*

Écrivez une fonction Python `retournement(tab, n)` qui prend en argument un tableau numpy 1D non vide de taille `n` et qui inverse l'ordre de ses éléments. La fonction ne renvoie rien, elle modifie juste le contenu du tableau passé en argument. Par exemple, si `tab` est le tableau

1	7	2	8
---	---	---	---

, après application de la fonction sur le tableau `tab`, ce dernier sera égal au tableau

8	2	7	1
---	---	---	---

.