

Examen du module d'ingénierie informatique

Durée : 1 heure 30

Documents autorisés : Aucun document.

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.

```
x = 5

def fonction(x) :
    x = x + 1

fonction(x)
print("valeur finale de x =", x)
```

Exercice 2 (2 points) — *Mirroir*

Écrivez en Python une fonction `mirroir(x)` qui, étant donné un nombre entier x compris entre 10 et 99 renvoie le nombre entier *mirroir* y tel que le chiffre des unités de y est égal au chiffre des dizaines de x et le chiffre des dizaines de y est égal au chiffre des unités de x . Par exemple `mirroir(24)` renverra le nombre 42.

Exercice 3 (3 points) — *Permutation circulaire*

Écrivez en Python une fonction `permutation(tab, n)` qui, étant donné un tableau numpy 1D `tab` contenant n éléments, **modifie** celui-ci de manière à réaliser une permutation circulaire d'un élément vers la droite. La fonction ne renvoie rien, elle modifie juste le tableau. Par exemple, l'application de la fonction sur le tableau `tab = [5 7 2 4]` modifiera celui-ci en `tab = [4 5 7 2]`.

Exercice 4 (3 points) — *Delta Min Max*

Écrivez en Python une fonction `delta(tab, nbLigs, nbCols)` qui, étant donné un tableau numpy 2D `tab` d'entiers de `nbLigs` lignes et `nbCols` colonnes, renvoie le nombre égal à la différence des entiers

maximum et minimum contenus dans le tableau. Par exemple, le tableau `tab =`

2	5	12	-4
4	-3	0	7
3	1	9	5

a pour valeur maximale 12 et valeur minimale -4 . Donc, pour ce tableau, la fonction renverrait la valeur $12 - (-4) = 16$. On supposera dans cet exercice que le tableau `tab` n'est jamais vide.

Exercice 5 (2 points) — Analyse

Écrivez en Python une fonction `analyse(x)` qui prend en argument un nombre x . Si celui-ci est inférieur strictement à 10, la fonction renvoie la valeur de $x - 1$. S'il est compris entre 10 (inclus) et 50 (exclus), elle renvoie la valeur de x . S'il est compris entre 50 (inclus) et 100 (exclus), elle renvoie la valeur de $x + 1$. Enfin, s'il est supérieur ou égal à 100, la fonction renvoie la chaîne de caractères « nombre trop grand ».

Exercice 6 (3 points) — Bon voisinage

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 7 (2 points) — La somme des deux

Écrivez une fonction Python `somme(tab1, tab2, nbLigs, nbCols)` qui, étant donné deux tableaux numpy 2D d'entiers `tab1` et `tab2` de `nbLigs` lignes et `nbCols` colonnes, renvoie un nouveau tableau numpy de même taille tel que, pour tout (x, y) , l'élément dans la cellule sur la colonne x et ligne y de ce nouveau tableau est égal à la somme des éléments de même coordonnées dans les tableaux `tab1` et

`tab2`. Par exemple, la somme de

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

 et

1	3	1	2
1	2	0	0
4	2	1	0

 produira le tableau

3	8	5	8
5	4	0	7
7	3	9	5

.

Exercice 8 (3 points) — Région englobante

Écrivez une fonction Python `region(tab, n)` qui prend en paramètre un tableau numpy 2D `tab` de n colonnes et 2 lignes. Chaque colonne représente les coordonnées (x, y) (en pixels) du coin supérieur gauche d'une image dans une fenêtre pyQt5. On suppose ici que $n > 0$. Toutes les images sont d'exactly 20x20 pixels. La fonction doit renvoyer un rectangle `QRect` de la région la plus petite contenant toutes les images de `tab`, comme le montre la figure 1. On rappelle que, pour créer un `QRect`, on utilise le constructeur `QRect(x, y, w, h)`, où (x, y) sont les coordonnées (en pixels) du coin supérieur gauche de la région, w est la largeur de la région (en pixels) et h sa hauteur (toujours en pixels).

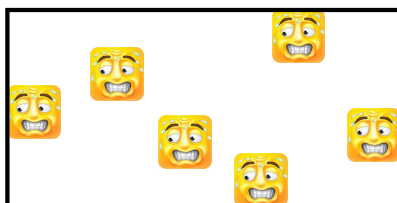


FIGURE 1 – Une région englobante (rectangle noir).