



Programmation impérative en Python – SPUF21

Année 2021-2022 – Examen terminal

Nom :

Prénom :

Numéro d'étudiant :

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

Durée : 2 heures.

Aucun document n'est autorisé. L'usage de la calculatrice ou de tout autre appareil électronique est interdit.

Les exercices sont indépendants. Au sein d'un même exercice, vous pouvez utiliser les variables et fonctions des questions précédentes, même si vous n'avez pas su les faire; chaque question est donc indépendante.

À part les méthodes et fonctions de base, vous n'avez pas le droit d'utiliser les fonctions et les méthodes « avancées », sauf si l'énoncé vous conseille l'utilisation de certaines d'entre elles.

```
1 # Fonctions autorisées
2 len(...)
3 range(...)
4 print(...)
5
6 # Méthodes autorisées
7 L.append(x) (ainsi que son équivalent pour les ensembles)
```

```
1 # Par exemple les méthodes et fonctions suivantes sont entre autres interdites
2 max(...) min(...) sum(...)
3 s.split(...) s.index(...) L.extend(...)
4
5 # Vous n'avez pas le droit d'utiliser des compréhensions ou des slices.
6 # À la place vous devez utiliser des boucles.
7 [ x for x in range(L) ]
8 chaine[début:fin:pas]
```




Exercice 1 Questions rapides 3 points

0 0,5 1 1,5 2 2,5 3

1. Construire avec une boucle une liste $L = [0.0, 0.5, \dots, 19.5, 20.0]$ contenant toutes les valeurs de 0 à 20 avec un pas de $\frac{1}{2}$.

.....
.....
.....
.....
.....

2. On se donne une chaîne s . En utilisant une boucle `for`, définir un ensemble E contenant tous les caractères de s . On n'aura évidemment pas le droit de faire $E = \text{set}(s)$.

.....
.....
.....
.....
.....

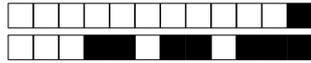
3. On définit ci-dessous une fonction `construire_matrice(n)` qui renvoie une matrice de n lignes et n colonnes.

```
1 def construire_matrice(n):  
2     ligne = [0] * n  
3     M = []  
4     for i in range(n):  
5         M.append(ligne)  
6     return M
```

En utilisant cette fonction `construire_matrice(n)` on initialise une matrice M et on la modifie. Que vaut M après la modification? Justifier.

```
1 >>> M = construire_matrice(3)  
2 >>> M  
3 [[0, 0, 0], [0, 0, 0], [0, 0, 0]]  
4 >>> M[0][0] = 5
```

.....
.....
.....
.....



Exercice 4 Des polynômes et des listes 5 points

0 0,5 1 1,5 2 2,5 3 3,5 4 4,5 5

Travailler sur des polynômes avec des arbres n'est guère pratique. On va alors représenter les polynômes avec des listes. Concrètement le polynôme $5+3x-12x^2+7x^3$ sera représenté par la liste $[5, 3, -12, 7]$.

1. Écrire une fonction **calcul**(P, x0) qui prend une liste P représentant un polynôme p et renvoie la valeur de $p(x_0)$. Ainsi calcul([3, 2, 5], 10) renverra $3 + 2x_0 + 5x_0^2 = 3 + 2 \times 10 + 5 \times 10^2 = 523$.

.....
.....
.....
.....
.....
.....
.....

2. Écrire une fonction **degré**(P) qui renvoie le degré du polynôme P, c'est-à-dire le plus grand indice i tel que P[i] soit non nul. Ainsi, degré([2, 0, 5, 4, 0, 0, 0]) renverra 3 (car ici, $P(x) = 2 + 5x^2 + 4x^3$). Si tous les coefficients sont nuls, on renverra alors 0.

.....
.....
.....
.....
.....
.....
.....

3. Écrire une fonction **réduire**(P) qui renvoie une nouvelle liste Q mais sans 0 inutiles à droite de la liste P. Par exemple réduire([2, 0, 5, 4, 0, 0, 0]) renverra [2, 0, 5, 4].

.....
.....
.....
.....
.....
.....
.....



Exercice 6 Élections 3 points

0 0,5 1 1,5 2 2,5 3

DANS UN PAYS FICTIF, des citoyens déçus par le résultat de l'élection présidentielle ont décidé de fuir dans l'arrière-pays de Nice pour y créer un groupe de résistance dans le maquis. Pour guider leur nouvelle commune anarchisante et auto-gérée, ils décident d'élire deux chefs sans pouvoir, le pouvoir étant réparti entre tous. Les bulletins de votes sont stockés sous forme de chaînes dans une liste de la forme : ['Honoré Leprof', 'Noé Meslacets', 'Enrichir Sescollègues', 'Honoré Leprof', ...]

1. Écrire une fonction `compter(liste)`, qui à partir d'une liste de noms renvoie le dictionnaire donnant le nombre de voix pour chacun de ces noms.

```
1 >>> voix = compter(liste) ; print(voix["Honoré Leprof"], voix["Thésée Vous"])
2 184 123
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

2. Écrire une fonction `gagnants(voix)` qui à partir du dictionnaire de la question précédente affiche les deux candidats avec le plus de voix (en cas d'égalité entre deux seconds, peu importe celui qu'on choisit). Pour récupérer deux noms quelconques, on pourra utiliser :

```
1 >>> tuple(voix)[0:2]
2 ('Thésée Vous', 'Pompée Félix')
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

