

Exercice 1

1. En voulant programmer une fonction qui calcule la valeur minimale d'une liste d'entiers, on a écrit :

```
def minimum(L):
    mini = 0
    for e in L:
        if e < mini:
            mini = e
    return mini
```

Cette fonction a été mal programmée. Pour quelle liste ne donnera-t-elle pas le résultat attendu, c'est-à-dire son minimum ?

- a) [-1, -8, 12, 2, 23]
- b) [0, 18, 12, 2, 3]
- c) [-1, -1, 12, 12, 23]
- d) [1, 8, 12, 2, 23]

2. La fonction maxi ci-dessous a pour but de renvoyer la valeur maximale présente dans la liste qui lui est passée en argument.

```
def maxi(L):
    dernier_indice = len(L) - 1
    valeur_max = L[0]
    for i in range(1, dernier_indice):
        if L[i] > valeur_max:
            valeur_max = L[i]
    return valeur_max
```

Cette fonction a été mal programmée. On souhaite réaliser un test pour le démontrer. Parmi les propositions suivantes, laquelle mettra la fonction maxi en défaut ?

- a) `assert maxi([1, 2, 3, 4]) == 4`
- b) `assert maxi([4, 3, 2, 1]) == 4`
- c) `assert maxi([1, 3, 3, 2]) == 3`
- d) `assert maxi([1, 1, 1, 1]) == 1`

3. On exécute le script suivant :

```
liste = [17, 12, 5, 18, 2, 7, 9, 15, 14, 20]
somme = 0
i = 0
while i < len(liste):
    somme = somme + liste[i]
    i = i + 1
resultat = somme / len(liste)
```

Quelle affirmation est fautive parmi les suivantes ?

- a) le corps de la boucle a été exécuté 10 fois
- b) à la fin de l'exécution la valeur de i est 9
- c) resultat contient la moyenne des éléments de liste
- d) len est une fonction

4. T est un tableau de nombres entiers non vide. Que représente la valeur de s renvoyée par cette fonction ?

```
def mystere(T):
    s = 0
    for k in T:
        if k % 2 == 0:
            s = s+k
    return s
```

- a) la somme des valeurs du tableau T
- b) la somme des valeurs positives du tableau T
- c) la somme des valeurs impaires du tableau T
- d) la somme des valeurs paires du tableau T

5. Lors de l'exécution du code suivant, combien de fois l'opération $a = 2^*a$ sera-t-elle effectuée ?

```
a = 1
cpt = 1
while cpt < 8:
    a = 2*a
    cpt = cpt+1
```

- a) 0
- b) 1
- c) 7
- d) 8

```
def produit (L):  
    p = ...  
    for elt in L:  
        .....  
    return p
```

6. La fonction suivante doit calculer le produit de tous les éléments de la liste passée en paramètre. Avec quelles expressions doit-on la compléter pour que cette fonction soit correcte?

- a) 1 puis $p = p * elt$
- b) 0 puis $p = p * elt$
- c) 1 puis $p = elt$
- d) 0 puis $p = elt$

7. Quelle valeur permet de compléter l'affirmation suivante : « Le nombre d'opérations nécessaires pour rechercher un élément séquentiellement (l'autre méthode que la recherche par dichotomie) dans un tableau de longueur n est de l'ordre de ... »?

- a) 1
- b) n
- c) n^2
- d) n^3

8. Une page HTML comporte ce bout de code :

```
<article>  
<h1>Les auteurs de ce site</h1>  
<button onclick="auteurs()">Appuyer ici</button>  
</article>  
  
<script src="java_script/ScriptJava.js"></script>  
<script> function auteurs() { alert("Auteurs anonymes"); } </script>
```

Quel sera le résultat d'un clic sur le bouton "Appuyer ici" ?

- a) Rien
- b) La page du navigateur se fermera
- c) La page affichera à la place du bouton "Appuyer ici" le message "Auteurs anonymes"
- d) Cela provoquera l'ouverture d'une fenêtre comportant le message "Auteurs anonymes"

9. Un élément form (un formulaire) d'une page HTML contient un élément button de type submit. Un clic sur ce bouton

- a) envoie les données du formulaire vers la page définie par l'attribut action de l'élément form
- b) efface les données entrées par l'utilisateur dans le formulaire
- c) envoie les données du formulaire vers la page définie par l'attribut method de l'élément form
- d) ne fait rien du tout si un script javascript n'est pas associé au bouton

10. Dans une page HTML, lequel de ces codes permet la présence d'une div qui appelle la fonction javascript spoil() lorsque l'utilisateur passe la souris dessus?

- a) `<div href="spoil()">Passe ta souris</div>`
- b) `<div if_mouseover="spoil()">Passe ta souris</div>`
- c) `<div value="Passe ta souris"><div> spoil()</div></div>`
- d) `<div onmouseover="spoil()">Passe ta souris ici</div>`

Dans le dossier partager, vous trouverez un dossier mon beau site. Copiez-le dans votre répertoire personnel.

1. Quels éléments HTML ont été associé à un évènement? Préciser le type d'évènement.
2. Modifier le comportement des boutons Bleu et Rouge pour que l'évènement soit activé au survole de la souris.
3. Ajoute un évènement au bouton A toi de jouer pour que sur un click de l'utilisateur, il soit écrit Bravo! dans le span en dessous.

Une fois votre travail terminé, vous ferez une archive avec les fichiers de votre page que vous nommerez nom-prenom-ex2.7z puis vous placerez cette archive dans le dossier devoir.

Exercice 3

Sous liste

Attention, cette exercice est long et des questions ne sont pas évidentes, il ne faut pas hésitez à les sauter car elles sont toutes indépendantes.

Étant donné un tableau non vide de nombres entiers relatifs, on appelle sous-séquence une suite non vide d'éléments voisins de ce tableau. On cherche dans cet exercice à déterminer la plus grande somme possible obtenue en additionnant les éléments d'une sous-séquence.

Par exemple, pour le tableau ci-dessous, la somme maximale vaut 18. Elle est obtenue en additionnant les éléments de la sous-séquence encadrée en gras ci-dessous (6;8;-6;10).

-8 -4 **6 8 -6 10** -4 -4

1. (a) Quelle est la solution du problème si les éléments du tableau sont tous positifs?
(b) Quelle est la solution du problème si tous les éléments sont négatifs?
2. Dans cette question, on examine toutes les sous-séquences possibles
 - (a) Écrire le code Python d'une fonction `somme_sous_sequence(lst, i, j)` qui prend en argument une liste et deux entiers `i, j` et renvoie la somme de la sous-séquence délimitée par les indices `i` et `j` (inclus).
 - (b) La fonction `pgsp` ci-dessous permet de déterminer la plus grande des sommes obtenues en additionnant les éléments de toutes les sous-séquences possibles du tableau `lst`

```
def pgsp(lst:list):
    n = len(lst)
    somme_max = lst[0]
    for i in range(n):
        for j in range(i, n):
            s = somme_sous_sequence(lst, i, j)
            if s > somme_max :
                somme_max = s
    return somme_max
```

Parmi les quatre choix suivants, quel est le nombre de comparaisons effectuées par cette fonction si le tableau `lst` passé en paramètre contient 10 éléments? Vous justifierez votre réponse.

10 55 100 1055

- (c) Expliquer pourquoi la complexité de cet algorithme est en $\mathcal{O}(n^2)$.
- (d) Recopier et modifier la fonction `pgsp` pour qu'elle renvoie un tuple contenant la somme maximale et les indices qui délimitent la sous-séquence correspondant à cette somme maximale.

3. Ici, on propose une solution plus astucieuse qui consiste à la liste des sommes maximales des sous séquences qui se terminent par les valeurs de la liste à étudier. On a programmé la fonction suivante

```
def pgsp2(lst:list):
    sommes_max = [lst[0]]
    for i in range(1, len(lst)):
        if lst[i] > (sommes_max[-1] + lst[i]):
            sommes_max.append(lst[i])
        else:
            sommes_max.append(sommes_max[-1] + lst[i])
    return maximum(sommes_max)
```

- (a) Écrire la fonction `maximum` qui prend en argument une liste et qui renvoie la valeur maximal de la liste.
- (b) Quelle est la complexité de votre fonction `maximum`?
- (c) Faire une estimation de la complexité de la fonction `pgsp2`. Que peut-on dire par rapport à la fonction `pgsp`?