

Interrogation d'Informatique n°2

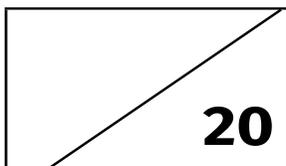
Lundi 08/01/2023
Durée : 30 minutes

Consignes

- En Mathématiques, les énoncés du cours doivent être complètement écrits : un cadre, des hypothèses et une conclusion.
- En Informatique : les scripts doivent être correctement indentés, en mettant en valeur l'indentation à l'aide d'une barre verticale.
- La note finale tiendra compte, directement ou indirectement, de la qualité de la rédaction et de la présentation.
- Le crayon à papier ne sera pas corrigé. ● L'usage de la calculatrice est interdit.

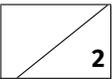
Nom :

Prénom :

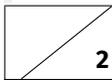


Exercice 1 | Scripts de cours.

[_____ / 4]

1.  Écrire une fonction réursive d'en tête factorielle(n) prenant en argument un entier naturel n et qui renvoie la factorielle de n.



2.  Écrire une fonction d'en-tête tri_test_decroissant(L), qui prend en paramètre une liste L et qui renvoie **True** si elle est triée par ordre décroissant, et **False** sinon.



Exercice 2 | Fonction mystère.

[_____ / 3]

On considère le programme ci-après.

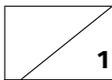
```
def mystere(n):
    if n == 0:
        return 1
    else:
        return n + mystere(n-2)
```

1.  Que renvoie mystere(0) ?



2.  Que renvoie mystere(4) ?



3.  Que dire de l'appel mystere(3) ? Expliquez.



Exercice 3 | Le tri à bulles

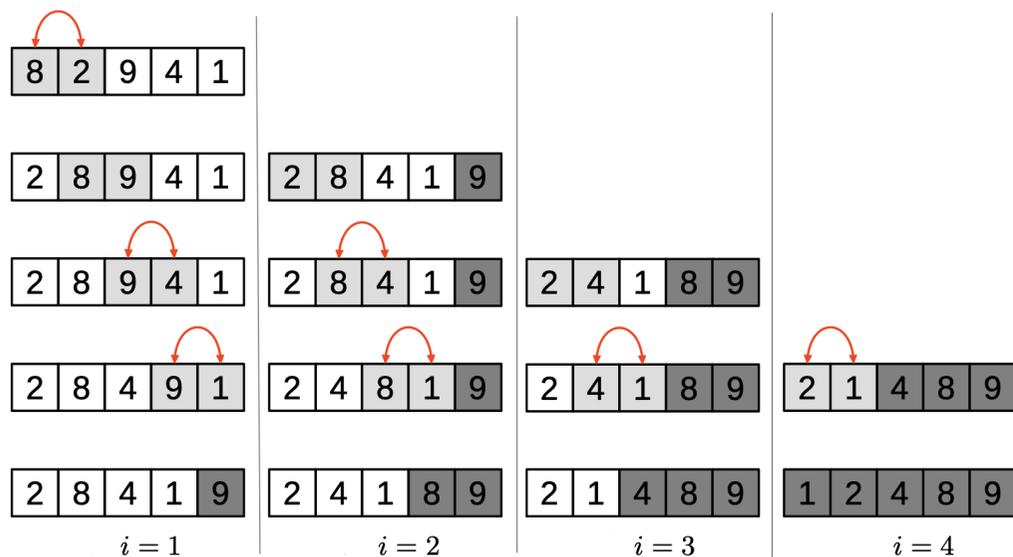
[_____ / 13]

Le tri à bulles est un algorithme de tri classique. Son principe est simple, et il est relativement facile à implémenter.

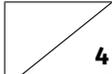
On considère une liste de nombres L .

- **[Étape (P)]** L'algorithme parcourt toute la liste, et dès que deux éléments consécutifs ne sont pas ordonnés dans l'ordre croissant, on les échange. Après un premier passage, le maximum de L sera alors « remonté » en fin de liste.
- On recommence ce parcours, tant que l'on a eu à effectuer un échange.

Au i -ème passage on fait remonter le i -ème plus grand élément du tableau à sa position définitive, un peu à la manière de bulles qu'on ferait remonter à la surface d'un liquide, d'où le nom d'algorithme de tri à bulles. Ci-dessous, vous trouverez un exemple d'exécution de l'algorithme de tri à bulles lorsque $L = [8, 2, 9, 4, 1]$. Les cases gris clair représentent les éléments comparés, les flèches rouges les échanges d'éléments, et les cases gris sombre les éléments placés définitivement.

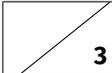


TRI À BULLES SUR UN EXEMPLE

1.  Appliquer l'algorithme de tri à bulles « à la main » à la liste ci-après, à la manière de l'exemple précédent. (on présentera le déroulé comme sur la figure précédente)

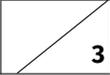
 $L = [5, 4, 1, 6]$

Le but de la suite de cet exercice est d'implémenter l'algorithme de tri à bulles.

2.  Compléter la fonction `parcours` d'argument une liste L et effectuant l'étape (P) du tri bulles, et qui renvoie en plus le nombre d'échanges effectués. Par exemple, si $L = [5, 4, 1, 6]$, alors `parcours(L)` renvoie 2 et la liste L sera modifiée en $L = [4, 1, 5, 6]$.

```
def parcours(L) :
    C = 0 # Compteur du nombre de permutations
    for i in range(_____):
        if _____:
            L[i], L[i+1] = L[i+1], L[i] # On échange
            _____ # On augmente C de 1
    return C
```

3.  On veut utiliser cette fonction `parcours` pour trier par ordre croissant une liste selon le principe suivant : pour une liste L donnée en entrée on répète l'application de la fonction `parcours` jusqu'à ce que L devienne invariante par cette fonction (autrement dit, jusqu'à ce que la fonction `parcours` retourne 0). Écrire une fonction `tri_bulles(L)` qui ordonne une liste L donnée en entrée selon ce principe. *Remarque : la fonction `tri_bulles` doit modifier directement la liste L , elle ne retourne aucune valeur.*

4.  Le tri à bulles est-il un tri récursif? Comparatif? En place? Justifier.

Interrogation d'Informatique n°2

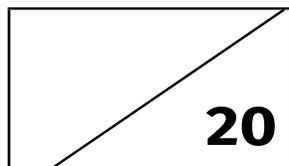
Lundi 08/01/2023
Durée : 30 minutes

Consignes

- En Mathématiques, les énoncés du cours doivent être complètement écrits : un cadre, des hypothèses et une conclusion.
- En Informatique : les scripts doivent être correctement indentés, en mettant en valeur l'indentation à l'aide d'une barre verticale.
- La note finale tiendra compte, directement ou indirectement, de la qualité de la rédaction et de la présentation.
- Le crayon à papier ne sera pas corrigé. ● L'usage de la calculatrice est interdit.

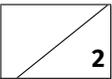
Nom :

Prénom :



Exercice 1 | Scripts de cours

[— / 4]

1.  Écrire une fonction réursive d'en tête factorielle(n) prenant en argument un entier naturel n et qui renvoie la factorielle de n.



2.



- Écrire une fonction d'en-tête `tri_test_croissant(L)`, qui prend en paramètre une liste L et qui renvoie **True** si elle est triée par ordre croissant, et **False** sinon.



Exercice 2 | Fonction mystère

[— / 3]

On considère le programme ci-après.

```
def mystere(n):
    if n == 1:
        return 0
    else:
        return n**2 + mystere(n-2)
```

1.



- Que renvoie `mystere(0)` ?



2.



- Que renvoie `mystere(4)` ?



3.



- Que dire de l'appel `mystere(3)` ? Expliquez.



Exercice 3 | Le tri à bulles

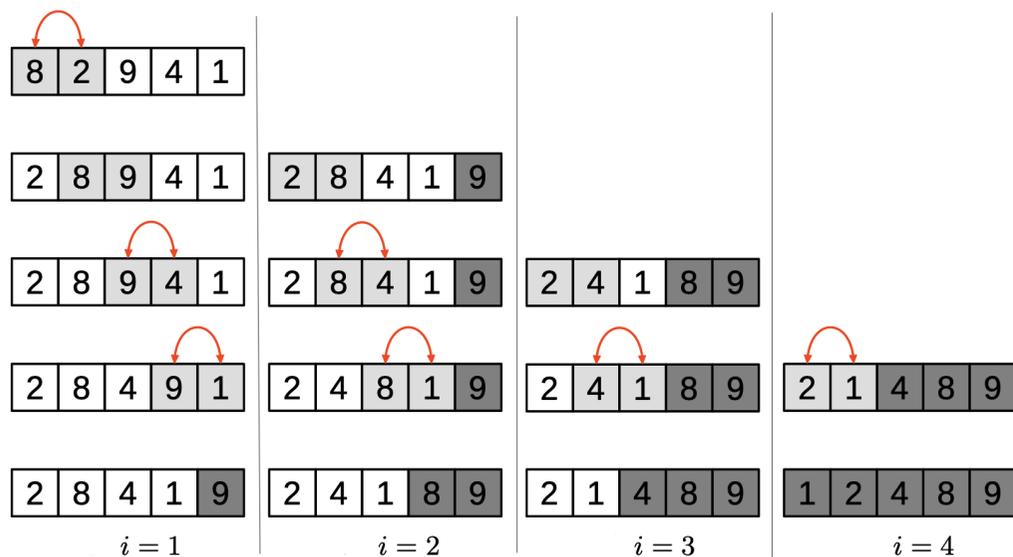
[_____ / 13]

Le tri à bulles est un algorithme de tri classique. Son principe est simple, et il est relativement facile à implémenter.

On considère une liste de nombres L .

- **[Étape (P)]** L'algorithme parcourt toute la liste, et dès que deux éléments consécutifs ne sont pas ordonnés dans l'ordre croissant, on les échange. Après un premier passage, le maximum de L sera alors « remonté » en fin de liste.
- On recommence ce parcours, tant que l'on a eu à effectuer un échange.

Au i -ème passage on fait remonter le i -ème plus grand élément du tableau à sa position définitive, un peu à la manière de bulles qu'on ferait remonter à la surface d'un liquide, d'où le nom d'algorithme de tri à bulles. Ci-dessous, vous trouverez un exemple d'exécution de l'algorithme de tri à bulles lorsque $L = [8, 2, 9, 4, 1]$. Les cases gris clair représentent les éléments comparés, les flèches rouges les échanges d'éléments, et les cases gris sombre les éléments placés définitivement.



TRI À BULLES SUR UN EXEMPLE

1.  Appliquer l'algorithme de tri à bulles « à la main » à la liste ci-après, à la manière de l'exemple précédent. (on présentera le déroulé comme sur la figure précédente)

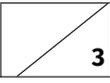
 $L = [5, 4, 1, 7]$

Le but de la suite de cet exercice est d'implémenter l'algorithme de tri à bulles.

2.  Compléter la fonction `parcours` d'argument une liste L et effectuant l'étape (P) du tri bulles, et qui renvoie en plus le nombre d'échanges effectués. Par exemple, si $L = [5, 4, 1, 7]$, alors `parcours(L)` renvoie 2 et la liste L sera modifiée en $L = [4, 1, 5, 7]$.

```
def parcours(L) :
    C = 0 # Compteur du nombre de permutations
    for i in range(_____):
        if _____:
            L[i], L[i+1] = L[i+1], L[i] # On échange
            _____ # On augmente C de 1
    return C
```

3.  On veut utiliser cette fonction `parcours` pour trier par ordre croissant une liste selon le principe suivant : pour une liste L donnée en entrée on répète l'application de la fonction `parcours` jusqu'à ce que L devienne invariante par cette fonction (autrement dit, jusqu'à ce que la fonction `parcours` retourne 0). Écrire une fonction `tri_bulles(L)` qui ordonne une liste L donnée en entrée selon ce principe. *Remarque : la fonction `tri_bulles` doit modifier directement la liste L , elle ne retourne aucune valeur.*

4.  Le tri à bulles est-il un tri récursif? Comparatif? En place? Justifier.