

Épreuve d'Informatique 2

Durée 1 h

L'usage des calculatrices est interdit.

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies. Les algorithmes doivent être commentés.

Exercice 1 (Insertion)

Écrire une fonction python effectuant un tri par insertion sur une liste L de valeurs numériques. On indiquera bien les indentations en tirant un trait.

Rappeler les complexités de ce tri dans les cas suivants : meilleurs, moyen et pire.

Exercice 2 (Euler)

Implémenter une résolution d'équation différentielle par la méthode d'Euler : la fonction `euler` prendra pour arguments a , b , F , y_0 et n , et résoudra l'équation différentielle $y' = F(x, y)$ sur l'intervalle $[a, b]$ avec n points et la condition initiale $y(a) = y_0$.

Exercice 3 (Algorithme à comprendre)

Voici le code d'une mystérieuse fonction, qui prend en argument une liste L et un entier naturel $i \in \mathbb{N}$ inférieur strictement à la taille de la liste :

```
1 def mystere(L, i):
2     if i == len(L) - 1:
3         return True
4     if L[i] > L[i+1]:
5         return False
6     return mystere(L, i+1)
```

- 1) Soit $L = [6, 9, 4, 8, 12]$. Que retournent `mystere(L, 2)` et `mystere(L, 0)` ? Donner la liste des appels récurrents dans chacun des cas.
- 2) Que fait cette fonction dans le cas général ?
- 3) Quel est le nombre maximum d'appels récurrents (en fonction de n et i) si la liste est de longueur n ?
- 4) Écrire une version non récurrente de cet algorithme.
- 5) En vous inspirant de la fonction `mystere`, écrire une fonction récurrente `est_dans(L, x, i)` qui retourne `True` si x apparaît dans L à partir de l'indice i , et `False` sinon.

Exercice 4

On appelle palindrome un mot qui peut se lire indifféremment de gauche à droite ou de droite à gauche. Exemples : radar, sagas, kayak...

- 1) Écrire une fonction récurrente `palindrome(chaine)` prenant en paramètre une chaîne de caractères et retournant `True` s'il s'agit d'un palindrome et `False` sinon.
- 2) Soit n la longueur du mot. Calculer en fonction de n , la complexité de la fonction `palindrome()` dans le pire des cas.

FIN DE L'ÉPREUVE