

# Épreuve d'Informatique 1

Durée 2 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

Écrire une fonction python effectuant un tri rapide, ou *quicksort*, 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

Écrire une fonction `dichotomie` qui prendra comme argument une liste  $L$  de valeurs numériques triées et un élément  $x$  et cherchera  $x$  dans  $L$ . Si  $x \in L$ , `dichotomie` retourne le plus grand indice de  $x$  dans  $L$ , et retourne  $-1$  sinon. L'algorithme utilisé sera une dichotomie.

## Exercice 3

Voici le code d'une mystérieuse fonction, qui prend en argument un entier naturel  $n \in \mathbb{N}$  :

```
1 def mystere(n):
2     resultat = ""
3     while n > 0:
4         resultat = str(n % 2) + resultat
5         n = n // 2
6     return resultat
```

- 1) Donner le résultat de `mystere(13)`. On précisera le type du résultat. Que fait cette fonction ?
- 2) Détecter un bug potentiel, proposer une façon de le réparer.
- 3) Écrire une version récursive de cet algorithme.
- 4) L'adapter au cas d'une base quelconque.

## Exercice 4 (Parenthésage)

On considère le problème suivant : étant donnée une chaîne de caractères, déterminer si elle est *bien parenthésée*. Une chaîne bien parenthésée est soit une chaîne sans parenthèses, soit la réunion de deux chaînes bien parenthésées, soit un chaîne bien parenthésée entre parenthèses.

Par exemple les chaînes suivantes sont bien parenthésées :

"(oui (mais non) mais si (enfin)) bref"    ""    "bla"    "()()"

Les chaînes suivantes ne le sont pas :

"()("    ")bla(blo)bli("

- 1) Proposer une fonction `test_par` qui prend en argument une chaîne `chaine` teste si elle est bien parenthésée.

- 2) Proposer une fonction `test_multi_par` qui, à l'aide d'une pile, teste si une chaîne `chaine` comportant un parenthésage sous forme de parenthèses ( ), de crochets [ ] et d'accolades { } est bien parenthésée. Cette fonction retourne aussi un booléen.

Exemple de chaîne mal parenthésée : "(bla{blo})"

### Exercice 5 (bonus : des bulles)

Le tri à bulles ou tri par propagation est un algorithme de tri qui consiste à faire remonter progressivement les plus grands éléments d'une liste, comme les bulles d'air remontent à la surface d'un liquide.

L'algorithme : la machine parcourt la liste, et compare les couples d'éléments successifs. Lorsque deux éléments successifs ne sont pas dans l'ordre croissant, ils sont échangés. Après chaque parcours complet de la liste, l'algorithme recommence l'opération. Lorsqu'aucun échange n'a lieu pendant un parcours, cela signifie que la liste est triée. On arrête alors l'algorithme

Programmez le tri à bulle.

**FIN DE L'ÉPREUVE**