

Devoir d'informatique numéro 1

Vous devez coder vos algorithmes sur machine et me les envoyer sous forme d'un seul fichier python, commençant par « DL1_ » suivi de votre nom en majuscule sans accents.

Toute fonction doit être commentée juste après le `def` pour expliquer ce qu'elle fait. Les noms de variables doivent être explicites (mais pas trop long non plus). (Pour les plus à l'aise, vous pouvez lire PEP8).

Exercice 1 (Piles)

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

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

Exercice 2 (Récursivité : Dichotomie)

Soit $f : [a, b] \rightarrow \mathbb{R}$ continue telle que $f(a)f(b) < 0$, représentée par `f` qui associe un float à un float.

Écrire une fonction récursive `dicho_rec` qui résout $f(x) = 0$ à la précision ε .

Exercice 3 (Récursivité : Remplissage d'un contour)

Reprendre le TP image de l'an dernier pour la syntaxe de manipulation des images. Il est nécessaire de sauvegarder sur votre disque l'image

http://www.normalesup.org/~dconduche/informatique/PT/Devoirs/DL1_2016_motif.png

accessible depuis la page PT – info : <http://www.normalesup.org/~dconduche/informatique/PT/>

Le but de cet exercice est de remplir une zone blanche de l'image, sans dépasser la ligne noire : remplir soit l'intérieur soit l'extérieur du contour.

- 1) Écrire une fonction `charge_image` qui charge l'image dans un tableau NumPy.
- 2) Bien que l'image soit petite, il faut augmenter le niveau de récursion autorisé par Python :

```
import sys
sys.setrecursionlimit (10000)
```

Écrire une fonction récursive `remplissage` qui prend en argument les coordonnées d'un pixel (supposé blanc) et qui remplit la zone blanche qui contient le pixel, de proche en proche : la fonction colorie le pixel de départ, puis regarde si les pixels adjacents sont blanc ou noir, colorie les pixels blancs, et ainsi de suite.