

Tu pars en thèse, c'est compliqué

12 janvier 2017

1 le background

Les bons éditeurs de texte du genre d'emacs savent vérifier si une imbrication de parenthèses, accolades, et crochets est correcte, à savoir si à un ouvrant correspond bien un fermant, à la bonne place. Par exemple, on ne peut pas utiliser un fermant d'un certain type derrière un ouvrant d'un autre type. Mais il existe d'autres contraintes. Si l'imbrication n'est pas correcte, l'éditeur vous affiche un message du style `mismatch parentheses`. Essayez par exemple de taper dans emacs :

`([{ }])` et vous verrez le message apparaître.

La question est : comment vérifie-t-il ?

2 l'énoncé

L'objectif est de réaliser une fonction qui effectue cette vérification sur une chaîne uniquement composée de parenthèses, crochets et accolades ouvrants et fermants. Pour cela, votre programme doit lire sur l'entrée standard :

1. une ligne contenant un entier N , donnant le nombre d'expressions à traiter.
2. N lignes, contenant une suite de parenthèses, crochets, accolades ouvrants/fermants.

Pour chaque expression, votre programme doit simplement afficher sur la sortie standard `yes` ou `no` en fonction du fait que l'expression est correctement construite ou non. La figure 1 donne un exemple d'entrée et de résultat attendu.

entrée	sortie
4	
<code>({{[(())]})</code>	<code>no</code>
<code>{(){{{[[]()}}{[]}}]()}}}</code>	<code>yes</code>
<code>[{}][[]{([] [])}]</code>	<code>no</code>
<code>(()({[[]([[]() []])])}{()})</code>	<code>yes</code>

TABLE 1 – Exemple d'entrée et la sortie attendue

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur <http://cours-info.iut-bm.univ-fcomte.fr> un article dans la section `hackaton` → édition 2017, portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code, ainsi que le fichier d'entrée donné ci-dessus.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.