

A part en thèse, c'est simple

12 janvier 2017

1 le background

Dans la plupart des langages informatiques, les parenthèses sont utilisées pour régler des problème de priorité entre opérations. Par exemple, dans une expression arithmétique, cela permet d'indiquer explicitement une priorité à une addition précédée d'une multiplication.

Bien entendu, une parenthèse ouvrante doit obligatoirement être contrebalancée par une fermante quelque part dans l'expression. Sinon, elle n'est pas correcte. Un compilateur se doit donc de vérifier cela.

2 l'énoncé

L'objectif est de réaliser une fonction qui vérifie si une chaîne uniquement composée de parenthèses ouvrantes et fermantes est correctement parenthésée. 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 ouvrantes/fermantes.

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

entrée	sortie
4	
(())	no
((()((()())()))	yes
()())(((()())((()))())	no
()(((()(((()())((()))())	yes

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.