

TD: piles et notation polonaise inverse:

La Notation Polonaise Inverse (NPI), ou notation post-fixée, est une manière d'écrire les expressions mathématiques en se passant des parenthèses. Elle a été introduite par le mathématicien polonais Jan Lucasiewicz dans les années 1920.

Le principe de cette méthode est de placer chaque opérateur juste après ses deux opérandes. L'expression $2 + 3$ devient en NPI $23+$.

Regardons maintenant comment peuvent s'écrire les opérations un peu plus complexes au moyen de cette notation:

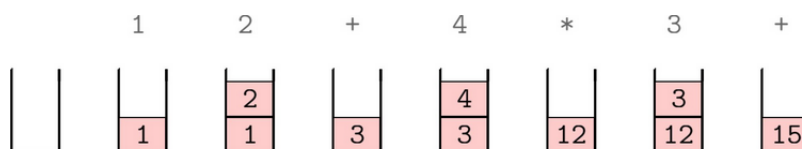
- $2 + 6 - 1$ s'écrit $26 + 1-$
- $5 * 3 + 4$ s'écrit $53 * 4+$
- $((1 + 2) * 4) + 3$ s'écrit $12 + 4 * 3+$

Évaluer une expression post-fixée est facile. Pour cela il suffit de lire l'expression de gauche à droite et d'appliquer chaque opérateur aux deux opérandes qui le précèdent. Si l'opérateur n'est pas le dernier symbole on replace le résultat intermédiaire dans l'expression et on recommence avec l'opérateur suivant.

Le but de l'exercice est de réaliser en Python une calculatrice simple, capable d'évaluer une formule en NPI et de retourner le résultat arithmétique. La réalisation d'une telle calculatrice se fera à l'aide d'une pile.

L'algorithme est très simple. On commence par lire un par un les caractères de l'expression. Si le caractère lu est un opérande alors on l'empile. Si le caractère lu est un opérateur, alors on dépile les deux éléments se trouvant en haut de la pile, on calcule le résultat en appliquant l'opérateur sur les deux opérandes dépilés et on empile le résultat. Une fois tous les caractères lus, la pile ne contient qu'un seul élément qui correspond au résultat final.

Voyons avec un exemple l'état de la pile après la lecture de chaque caractère de l'expression $((1 + 2) * 4) + 3$, ou $12 + 4 * 3+$ en NPI.



1 2 + 4 * 3 +

Vous pouvez remarquer que le résultat final 15 se trouve au sommet de la pile après la fin du programme.

Écrivez maintenant un programme Python qui met en oeuvre tout cela. Les caractères autorisés sont les chiffres de 0 à 9, ainsi que les symboles +, -, *, / correspondant aux 4 opérations élémentaires.

Votre programme pourra se composer des fonctions suivantes:

1. Une fonction `estOperateur(c)` qui prend en entrée un caractère et renvoie `True` s'il s'agit d'un opérateur et `False` sinon.
2. Une fonction `calcul(op, n, m)` qui prend en entrée un opérateur `op` parmi les quatre opérateurs autorisés et deux entiers `n` et `m` et qui renvoie le résultat du calcul `n op m`.
3. La fonction `evaluation(s)` qui prend en entrée une expression sous-forme de chaîne de caractères en notation polonaise inverse et renvoie le résultat du calcul.
4. Testez votre programme pour le calcul de l'expression $5*(8-3)*3+((3-1)*2)/3$ dont l'écriture en NPI est `583- *3 * 31 - 2 * 3/+`.