

```
public static void main (String [] args) {
    Pile p = empty ();
    push(4,p);
    push(5,p);
    System.out.println(top(p));
    pop(p);
    System.out.println(testempty(p));
    System.out.println(top(p));}
```

qui affiche 5, false, puis 4.

Exercice 6.12

La notation *postfixe* est une notation pour les expressions arithmétiques dans laquelle on écrit $3\ 4\ +$ l'expression habituellement écrite $3 + 4$: on écrit les deux arguments d'une fonction avant la fonction elle-même. Un avantage de cette notation est qu'elle ne demande pas de parenthèses, par exemple l'expression $(3 + 4) + 5$ est notée $3\ 4\ +\ 5\ +$, alors que l'expression $3 + (4 + 5)$ est notée $3\ 4\ 5\ +\ +$.

1. Comment s'écrit l'expression $3 + (4 + (5 + (6 + (7 + (8 + 9)))))$ en notation postfixe ?
2. On considère une expression telle que $3\ 4\ 5\ 6\ 7\ 8\ 9\ +\ +\ +\ +\ +\ +$ comme une liste d'éléments, treize dans cet exemple, telle que chaque élément est ou bien une constante ou bien le symbole d'une opération arithmétique. Définir un type pour les expressions arithmétiques postfixes.
3. On évalue une expression arithmétique postfixe en utilisant une pile. On lit les éléments de l'expression de la gauche vers la droite. Quand on lit une constante, on l'ajoute au sommet de la pile. Quand on lit le symbole de l'addition — resp. la soustraction, la multiplication, ... — on lit et supprime le sommet de la pile deux fois et on ajoute à la pile un nouvel élément qui est la somme — resp. la différence, le produit, ... — des deux éléments lus. Montrer que la lecture d'une expression complète ajoute à la pile un élément unique qui est la valeur de cette expression.
4. Programmer en Java une fonction qui lit une expression postfixe et calcule sa valeur.

Exercice 6.13 (L'élimination de la récursivité)

Un programme récursif peut toujours se transformer en un programme non récursif en utilisant une pile. À titre d'exemple, nous transformons dans cet