

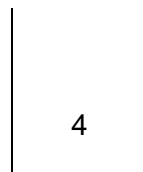
## I2181A : Langage C - Modularisation (TP3)

### Les structures récursives

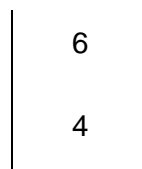
Ecrivez un programme `polonaaiseInverse` pour évaluer le résultat d'une expression arithmétique écrite en notation polonaise inverse. Pareille évaluation se fait aisément en utilisant une pile. Il faut parcourir l'expression : lorsqu'un nombre est rencontré, il est mis sur la pile ; lorsqu'un opérateur est rencontré, il faut dépiler deux nombres, calculer le résultat de l'opération entre le premier opérande (sous le sommet) et le second opérande (au sommet) et ensuite empiler le résultat.

Par exemple :  $4\ 6\ +\ 3\ *\ 8\ 3\ -\ /\$

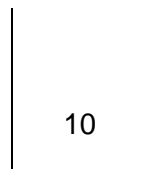
4 est mis sur la pile



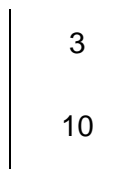
6 est ajouté au sommet de la pile



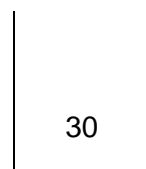
+ provoque le dépilage de 6 puis de 4 et la somme de 6 et 4 est mise sur la pile



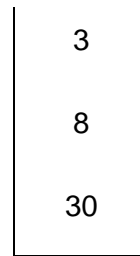
3 est ajouté au sommet de la pile



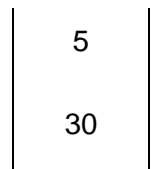
\* provoque le dépilage de 3 puis de 10 et le produit de 10 par 3 est mis sur la pile



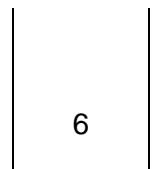
8 est ajouté au sommet de la pile, puis 3



- provoque le dépileage de 3 puis de 8 et la soustraction de 8 et 3 est mise sur la pile



/provoque le dépileage de 5 puis de 30 et le quotient, résultat de 30/5 est mis sur la pile



L'expression a été entièrement prise en compte, le résultat est le nombre qui reste dans la pile.

Voici l'interface d'une pile :

<code>File initPile ();</code>	qui renvoie une pile vide
<code>bool pileVide (File);</code>	qui teste si la pile est vide
<code>bool push (File*, int);</code>	qui renvoie vrai si l'entier a été placé sur la pile
<code>int pop (File*);</code>	qui retire l'entier du sommet de la pile et le renvoie
<code>void viderPile (File*);</code>	qui vide entièrement une pile
<code>void afficherPile (File);</code>	qui affiche le contenu de la pile

Schéma de la pile quand elle est vide :

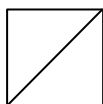
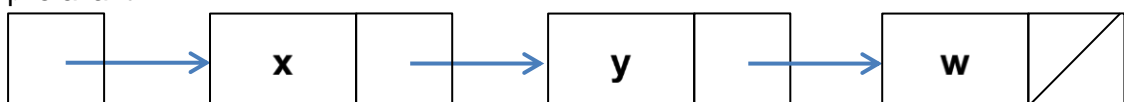


Schéma de l'ajout d'un élément au début (sommet) de la pile:

La pile avant



L'élément à rajouter



L'ajout

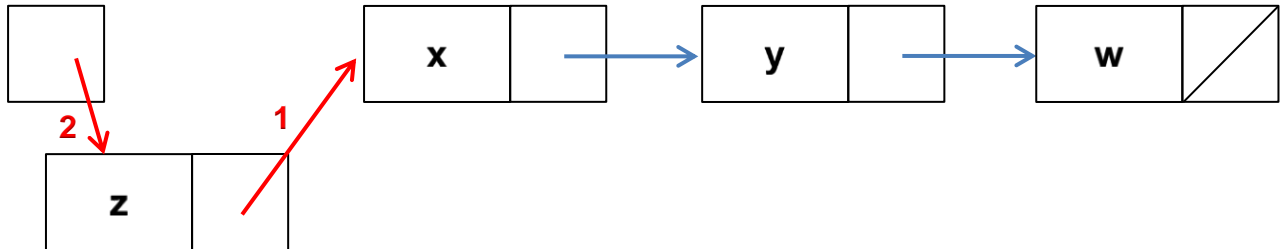
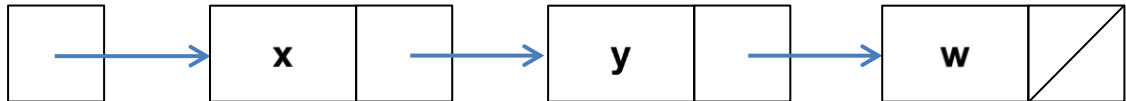
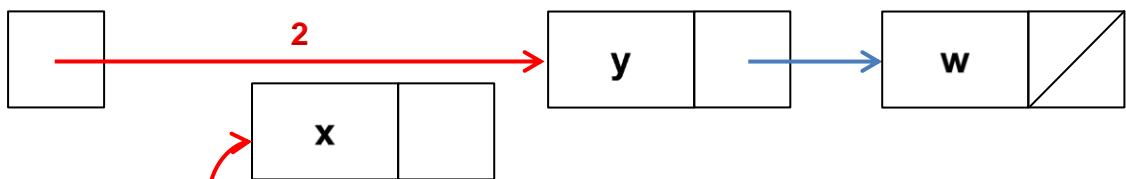


Schéma du retrait de l'élément du sommet de la pile:

La pile avant



Le retrait



L'élément retiré

1. Commencez par écrire un module *pile*. Celui-ci contiendra les fonctions nécessaires à l'implémentation d'une pile d'entiers à l'aide d'une liste chaînée, en respectant l'interface fournie.
2. Testez votre implémentation avec un petit programme qui ajoute (*push*) et retire (*pop*) des éléments à une pile et affiche son contenu après chaque opération. Pour ce faire, écrivez un *makefile*. Utilisez *valgrind* pour vérifier si votre module gère correctement la mémoire.
3. Ecrivez ensuite un programme qui va implémenter l'interpréteur d'expressions en notation polonaise inverse. Ce programme doit :
  - lire à l'entrée standard des lignes d'au plus 256 caractères qui contiennent chacune une expression arithmétique en notation polonaise inverse ; pour simplifier le code, nous supposons que les nombres sont des entiers composés d'un seul chiffre et les opérateurs arithmétiques sont limités aux 4 suivants : +, -, \*, /. Vous pouvez dans un premier temps vous limiter aux opérateurs d'addition et de soustraction.

- vérifier que la ligne lue ne contient que les caractères autorisés grâce à la fonction `strspn`
- évaluer l'expression lue
- afficher le résultat.

Vous pouvez utiliser les fonctions fournies par le module *utils* fourni.

Testez votre programme avec le fichier `npi.txt` comme suit :

```
./npi < npi.txt
```

Voici un exemple d'exécution avec ce fichier de test :

```
= manque des operandes: pas de reponse
1 2 + = la reponse est : 3
4 5 + 9 - 7 * = la reponse est : 0
4 8 + 2 - 0 / = division par 0
5 4 + = la reponse est : 9
4 8 + 2 - 0 / 9 + = division par 0
6 2 / = la reponse est : 3
4 8 + 2 - 0 9 + = trop d'operandes: expression incorrecte
3 3 * = la reponse est : 9
4 5 6 = trop d'operandes: expression incorrecte
4 2 * = la reponse est : 8
1 2 a 3 + + = Expression incorrecte en a
8 4 / = la reponse est : 2
4 5 6 - - - - = pile vide: pas de première opérande...
9 2 + = la reponse est : 11
- 4 5 = pile vide: pas de seconde opérande...
5 3 * = la reponse est : 15
4 - 5 = pile vide: pas de première opérande...
8 8 + = la reponse est : 16
```

Assurez-vous que votre programme ne provoque pas des fuites de mémoires à l'aide de `valgrind`.