

# INF304 — TP4

## Implémentation d'une calculette à l'aide d'une pile

Placez-vous dans votre répertoire INF304 et copiez les fichiers nécessaires à ce TP :

```
cp -r /Public/304_INF_Public/TP4 .
```

### Pile

#### ► Exercice 1

Prenez connaissance des fichiers suivants :

- `commandes_pile.c` et `commandes_pile.h` : paquetage de définition de commandes et de la lecture de commandes à partir d'un fichier. Le format du fichier est décrit dans le fichier *spécification*.
- `exemple_commandes_pile` : exemple de fichier de commandes de pile.
- `test_commandes_pile.c` : programme de test du paquetage **commandes\_pile**

Compilez et exécutez ce programme.

#### ► Exercice 2

Ecrire le fichier *corps* du paquetage **type\_pile** suivant les spécifications fournies dans le fichier `type_pile.h`.

Utiliser ensuite le programme principal `test_type_pile_entiers.c` :

- en argument, on fournit le nom d'un fichier de commande de pile,
- sur le modèle du programme **test\_commandes\_pile**, la suite de commandes est lue, et pour chacune d'elle la routine correspondante du paquetage **type\_pile** est appelée.

Créez des jeux de données afin de tester toutes les fonctions et procédures du paquetage.

### Calculette

Une fois votre paquetage **type\_pile** écrit, vous allez l'utiliser afin de simuler une calculette basique fonctionnant uniquement en nombres entiers relatifs (`int`) et utilisant uniquement les quatre opérations arithmétiques :

ADDITION + , SOUSTRACTION - , MULTIPLICATION \* ,  
DIVISION /

La calculette doit fonctionner en notation **post-fixée** (ou *polonaise inverse*) évitant ainsi l'usage des parenthèses : lors de la lecture des commandes, chaque valeur entière est **empilée**, et pour chaque opérateur, l'opération correspondante est effectuée sur les **deux valeurs dépilées** et le résultat est **empilé**.

Par exemple, le tableau ci-dessous donne un exemple d'évaluation de deux expressions

Notation infixée	( 5 - ( 8 / 4 ) ) * 6    vider_la_pile    (-4 - 7 ) * 2
Notation postfixée	5   8   4   /   -   6   *   vider_la_pile   -4   7   -   2   *

Ce qui donne le fichier de commande `exemple_commande_calculatrice`

Fichier de commandes	Action correspondante	État de la pile
r	Réinitialiser	[ ]
5	Empiler 5	[ 5 ]
8	Empiler 8	[ 5 8 ]
4	Empiler 4	[ 5 8 4 ]
/	Effectuer 8/4	[ 5 2 ]
-	Effectuer 5-2	[ 3 ]
6	Empiler 6	[ 3 6 ]
*	Effectuer 3*6	[ 18 ]
r	Réinitialiser	[ ]
-4	Empiler -4	[ -4 ]
7	Empiler 7	[ -4 7 ]
-	Effectuer -4-7	[ -11 ]
22	Empiler 22	[ -11 22 ]
d	Supprimer 22	[ -11 ]
2	Empiler 2	[ -11 2 ]

*	Effectuer $-11*2$	[ -22 ]
---	-------------------	---------

### ► Exercice 3

Prenez connaissance des fichiers suivants :

- `commandes_calculatrice.c` et `commandes_calculatrice.h` : paquetage de définition de commandes et de la lecture de commandes à partir d'un fichier. Le format du fichier est décrit dans le fichier *spécification*.
- `exemple_commandes_calculatrice` : exemple de fichier de commandes de calculatrice.
- `test_commandes_calculatrice.c` : programme de test du paquetage **`commandes_calculatrice`**

Compilez et exécutez ce programme.

### ► Exercice 4

En reprenant les canevas des programmes `test_commandes_calculatrice.c` et `test_type_pile_entiers.c`, écrire un programme `calculatrice.c` afin qu'il effectue les actions suivantes :

- le programme doit être appelé avec un argument correspondant au nom d'un fichier de commandes (dans le cas d'un nombre d'arguments différent de 1, un message d'erreur est affiché) ;
- le programme crée une pile vide pour la calculatrice ;
- le fichier est lu et les commandes sont effectuées une à une, le programme s'arrête lorsque toutes les commandes ont été lues et effectuées.

### Compte rendu :

À l'issue du TP, déposer sur [la page moodle du cours](#) :

- Un compte-rendu d'une page maximum, au format PDF, décrivant le jeu de données écrit permettant de couvrir l'ensemble des procédures du paquetage **`type_pile`**
- Le fichier `type_pile.c`
- Le code source de votre programme `calculatrice.c`