

## DM n° 1 — La PileMixte

Antoine de ROQUEMAUREL (Groupe 2.2)

---

### 1 Compilation

- `make` Compile les différents fichiers et produit l'exécutable de tests
- `make clean` supprime les fichiers binaires (.o)
- `make all` Recompile le projet après avoir supprimés tous les fichiers binaires

Les différents tests sont situés dans le fichier `./tests.c` dans 3 fonctions différentes.

- `testPileStatique`
- `testPileMixte`
- `testPileDynamique`

Pour lancer les tests, il faut lancer le fichier exécutable `./tests`, ceux-ci afficheront «Passed» si le test a réussi et «Failed» si celui-ci a échoué. (Si le test ne retourne pas la valeur attendue.).

Certains tests sont en commentaire, c'est parce que ce sont des tests qui fonctionnent dans un état anormal et doivent couper le programme à l'aide d'un `assert`. Afin d'éviter les problèmes dans le programme de tests, ceux-ci sont commentés mais peuvent être exécutés afin de vérifier que le programme s'arrête bien.

Les tests effectués sont les mêmes pour les trois piles, afin de pouvoir remarquer que la pile s'utilise de la même manière indépendamment de l'implémentation de la structure de donnée.

### 2 Petite modification au niveau de la spécification

Afin de pouvoir effectuer ce DM, j'ai dû modifier la spécification qui nous était donnée pour `empiler` et `depiler`, cette modification est présente dans les trois modules afin de garder une cohérence, bien que celle-ci n'était pas utile pour le module `pileStatique`.

En effet, la spécification demandée était la suivante :

```
void pileDynamiqueEmpiler(PileDynamique pPile, Element pElement);  
Element pileDynamiqueDepiler(PileDynamique pPile);
```

Cependant, lorsque l'on empile ou dépile, le pointeur de début de pile est modifié, ainsi en gardant la spécification demandée, la pile ne pouvait pas être modifiée lorsque l'on empilait ou dépilait.

Deux solutions se présentaient à moi pour palier à ce problème :

- Changer le prototype afin de retourner une nouvelle Pile
- Changer le premier paramètre afin d'y faire passer la pile via un pointeur

Afin de garder une cohérence avec les autres fonctions du module, j'ai choisi la seconde solution, ainsi les prototypes des fonctions `empiler` et `depiler` sont devenus ceux-ci :

```
void pileDynamiqueEmpiler(PileDynamique* pPile, Element pElement);  
Element pileDynamiqueDepiler(PileDynamique* pPile);
```

### 3 État incohérent du programme

Afin d'être sûr que le programme n'entre pas dans un état incohérent, en cas de problème le programme s'arrête à l'aide de `assert`. En effet, je pars du principe que l'utilisateur de la pile connaît les pré-conditions et ne doit pas essayer de dépiler une pile vide ou d'empiler si la pile est pleine. Si cela se produit, le programme s'arrêtera et il se rencontrera de lui même qu'il à effectué une erreur.

De plus, la fonction `depiler` doit renvoyer l'élément courant, cependant si la pile est vide, renvoyer la valeur est impossible.

### 4 Exemple d'exécution des tests

```
1 | aroquemaurel@Garp:~/cours/L2/sd4/DM
2 | $ ./tests
3 | [Passed] PileStatique  Création de la pile
4 | [Passed] PileStatique  Sommet
5 | [Passed] PileStatique  Empiler un élément
6 | [Passed] PileStatique  Depiler un élément
7 | [Passed] PileStatique  La pile est vide
8 | [Passed] PileStatique  La pile est pleine
9 | [Passed] PileStatique  On n'empile pas si la pile est pleine
10 | [Passed] PileStatique  La pile est vide
11 |
12 | [Passed] PileMixte  Création de la pile
13 | [Passed] PileMixte  Sommet
14 | [Passed] PileMixte  Empiler un élément
15 | [Passed] PileMixte  Depiler un élément
16 | [Passed] PileMixte  La pile est vide
17 | [Passed] PileMixte  Pile est pleine
18 | [Passed] PileMixte  On n'empile pas si la pile est pleine
19 | [Passed] PileMixte  La pile est vide
20 | [Passed] PileMixte  On ne dépile pas si la pile est vide
21 |
22 | [Passed] PileDynamique  Création de la pile
23 | [Passed] PileDynamique  Sommet
24 | [Passed] PileDynamique  Empiler un élément
25 | [Passed] PileDynamique  Depiler un élément
26 | [Passed] PileDynamique  La pile est vide
27 | [Passed] PileDynamique  Pile est pleine
28 | [Passed] PileDynamique  On n'empile pas si la pile est pleine
29 | [Passed] PileDynamique  La pile est vide
30 | [Passed] PileDynamique  On ne dépile pas si la pile est vide
```

Listing 1 – Exemple d'exécution des tests