

1 Réalisation d'une pile

Ce TP se déroule sur une séance. On s'intéresse au programme principal `main.c` suivant.

```
#include <stdio.h>
#include "pile.h"

int main ()
{   struct pile P;
    int i;

    init_pile (&P);
    for (i = 1; i < 10; i++)
        empiler (&P, i);
    while (! est_vide_pile (P))
    {   i = depiler (&P);
        printf ("%d\n", i);
    }
    clear_pile (&P);
    return 0;
}
```

Question 1. Sachant que P est une pile d'entiers, qu'est-ce que ce programme est censé afficher sur la sortie standard ?

On souhaite réaliser un module de pile d'entiers en utilisant le module de liste de doubles, étudié au premier TP. À la fin du TP, vous devrez donc avoir 5 fichiers : `main.c`, `pile.c`, `pile.h`, `liste_double.c` et `liste_double.h`. On impose le fichier `pile.h` suivant.

```
#include <stdbool.h>
#include "liste_double.h"

struct pile
{   struct liste_double L;
};

extern void init_pile (struct pile*);
extern void vider_la_pile (struct pile*);
extern void clear_pile (struct pile*);
extern void empiler (struct pile*, int);
extern int depiler (struct pile*);
extern bool est_vide_pile (struct pile);
```

Question 2. Écrire `pile.c`. Compiler et tester. Utiliser un *Makefile* (voir sur le site du cours). Vérifier avec `valgrind` que le programme est correct.

2 Réalisation d'une file

On s'intéresse au programme principal suivant.

```
#include <stdio.h>
#include "file.h"

int main ()
{   struct file F;
    int i;

    init_file (&F);
    for (i = 1; i < 10; i++)
        enfiler (&F, i);
    while (! est_vide_file (F))
    {   i = defiler (&F);
        printf ("%d\n", i);
    }
    clear_file (&F);
    return 0;
}
```

Question 3. Sachant que P est une file d'entiers, qu'est-ce que ce programme est censé afficher sur la sortie standard ?

On impose le fichier `file.h` suivant.

```
#include <stdbool.h>
#include "liste_double.h"

struct file
{   struct liste_double L;
};

extern void init_file (struct file*);
extern void vider_la_file (struct file*);
extern void clear_file (struct file*);
extern void enfiler (struct file*, int);
extern int defiler (struct file*);
extern bool est_vide_file (struct file);
```

Question 4. Écrire `file.c`. Quelles sont la ou les fonctions de manipulation de listes, qui seraient souhaitables pour réaliser une file, mais qui ne sont pas implantées ?

Question 5. Compléter le module `liste_double`. Que faut-il ajouter à `liste_double.h` ? à `liste_double.c` ?

Question 6. Suggérez une variante d'implantation des listes qui optimise le fonctionnement du module de file.