

TD4 Gestion de projet avec l'outil make

→ Exercice

Le but de cet exercice est de créer un exécutable appelée *moyenne* dont le rôle est de calculer la moyenne d'une suite de nombres saisis par l'utilisateur. Le code source est réparti dans cinq fichiers :

- *affichage.c* : partie du code réalisant l'affichage.
- *calcul.c* : partie du code exécutant le calcul.
- *entree.c* : partie du code en charge de la saisie des données.
- *principal.c* : programme principal, contenant la fonction principale *main*.
- *moyenne.h* : définition de la constante donnant le nombre de valeur à saisir.

On donne ci-dessous le squelette du fichier source *principal.c*, disponible sur l'ENT :

```
#include<stdio.h>
#include<stdlib.h>
...
/* prototypes des fonctions externes */
...

int main()
{
    float valeur = 0, chiffre;
    int i;

    printf("Nombre de valeurs : %d\n", NBREVALEUR);

    for(i=0; i<NBREVALEUR; i++)
    {
        /* saisie() à définir dans entree.c */
        chiffre = saisie();

        /* total() à définir dans calcul.c */
        valeur = total(valeur, chiffre);
    }

    /* affiche() à définir dans affichage.c */
    affiche(valeur);

    exit(0);
}
```

On donne également le contenu du fichier d'en-tête *moyenne.h* :

```
#define NBREVALEUR 5
```

1. Compléter la partie ❶ du programme (inclusion du fichier *moyenne.h*)
2. Compléter la partie ❷ du programme pour indiquer les prototypes des fonctions externes.

*En effet, on fait face ici à une problématique de compilation séparée : le code source est réparti dans plusieurs fichiers. De ce fait, pour utiliser dans un fichier source **F1.c** une fonction dont le code est contenu dans un autre fichier source **F2.c**, il faut spécifier dans le code du programme appelant **F1.c** le prototype de cette fonction.*

3. Rédiger le fichier source *affichage.c*.
4. Rédiger le fichier source *calcul.c*.
5. Rédiger le fichier source *entree.c*.
6. Donner l'arbre des dépendances correspondant à cet exécutable.

7. Rédiger le fichier *makefile* issu de l'arbre des dépendances. Compiler et tester.
8. A ce niveau, le projet présente un inconvénient : les fichiers intermédiaires tels que les modules objet restent sur le disque dur, même lors de la mise en production. Pour éviter cela, on peut ajouter dans le *makefile* une règle *clean* qui provoquera la suppression de ces modules objet. Cette suppression implique par ailleurs une régénération complète du projet si une nouvelle commande *make moyenne* est invoquée.

clean:

```
rm -rf *.o
```

Tester avec la commande *make clean*