

TP6

Exercice 1:

- 1) Proposer une solution du TP5 sous forme de module :
 - Partie interface : *ensemble.h*
 - Partie implémentation : *ensemble.c*
- 2) Proposer un programme test du module ensemble.h
- 3) Créer un *makefile* permettant de tester votre programme test.

Exercice 2:

La structure de données PILE peut être matérialisé sous forme d'un objet abstrait en adoptant une représentation contiguë. Compléter la partie implémentation correspondante à l'interface suivante (**n'oublier pas d'utiliser la technique des assertions**) :

Interface: Pile.h

```
void creepile (void) ;  
unsigned vide (void) ;  
int dernier (void) ;  
void empiler (int) ;  
void depiler (void) ;  
unsigned pleine (void) ;
```

Réalisation ou implémentation: Pile.c

```
#include<assert.h>  
#include "pile.h"  
/* représentation physique */  
# define n 100  
struct pile  
{  
    // à compléter  
};  
static struct pile p ;  
void creepile(void) { // à compléter }  
unsigned vide(void) {// à compléter }  
int dernier(void) {// à compléter }  
void empiler(int info) {// à compléter }  
void depiler(void){ // à compléter }  
unsigned pleine (void) {// à compléter }
```

Exercice 3 :

Enrichir l'objet abstrait PILE, concrétisé par une représentation contiguë, en intégrant les opérations suivantes :

- **nb_element** : renvoie le nombre d'éléments de la pile.
- **remplace_sommet** : change le sommet de la pile. Elle exige que la pile soit non vide.
- **effacer** : efface tous les éléments de la pile.

Exercice 4 :

Ecrire un programme en C permettant de lire un entier non signé et d'afficher tous les chiffres qui composent le nombre.

Indication: Envisager de réutiliser le module spécifique à la structure de données Pile, proposé dans l'exercice 1.

Exemple :

Si le nombre proposé est 2345 alors le programme souhaité doit afficher dans l'ordre: