

Série de TP N°03

---

Nous souhaitons écrire un programme en langage C pour réaliser quelques traitements sur une pile de valeurs entières. Le programme affichera le menu ci-dessous

```
#####  
1. Creer une pile d'entiers.  
2. Trier la pile  
3. Supprimer une valeur  
4. Supprimer les doublants  
5. Afficher le nombre d'occurrence d'une valeur  
6. Afficher la pile.  
7. Quitter.  
#####  
Votre choix :
```

**Travail à faire :**

Pour chacun des choix du menu, écrire une fonction langage C permettant de réaliser le traitement choisi, à savoir :

1. Une fonction *creerPile* permettant de créer une pile de valeurs entière. Lors de la création d'une pile, l'ancienne pile sera remplacée par la nouvelle. Alors, la fonction doit avertir l'utilisateur et il doit pouvoir annuler son choix.
2. Une fonction *trier* permettant de trier la pile dans le sens croissant ou décroissant de ses valeurs selon le choix de l'utilisateur.
3. Une fonction *supprimer* qui supprime toute les occurrences d'une valeur donnée par l'utilisateur.
4. Une fonction *supDoublant* qui supprime les valeurs répétitives dans la pile, on ne garde qu'une seule occurrence par valeur.
5. Une fonction *numOcc* qui calcule et affiche le nombre d'occurrences d'une valeur donnée par l'utilisateur.

**Remarque :**

- Selon le choix de l'utilisateur, le programme appellera la fonction correspondante.
- Le programme langage C permettant d'afficher le menu est disponible sur le site e-learning et ainsi qu'au verso de la série, le code inclue l'implémentation des fonctions élémentaires pour la gestion d'une pile : *empiler*, *depiler*, *sommet*, *viderPile* et ainsi qu'une fonction *AfficherPile* qui affiche le contenu de la pile.

Série de TP N°03

```
#include <stdio.h>
#include <stdlib.h>
typedef struct{
int valeur;
struct element *suivant;
}element;
void empiler(element **t,int val){
element *p = malloc(sizeof(element));
p->valeur = val;
p->suivant = *t;
*t = p;
}
void depiler(element **t){
element *p = *t;
if(p!= NULL)
*t = p->suivant;
free(p);
}
int sommet(element *t){
if(t!= NULL)
return t->valeur;
else{
puts("!!!! pile vide !!!!");
return 0;
}}
void viderPile(element **t){
element *q,*p = *t;
while(p!= NULL){
p = p->suivant;
free(q);
}
*t =NULL;
}
void AfficherPile(element *p){
element *q;
q = p;
while(q!=NULL){
printf("Val = %d \n",q->valeur);
q = q->suivant;
}}

int main(){
char c='1';
while(c!='o'){
printf("\n\n\n\n");
puts(" #####");
puts("1. Creer une pile d'entiers.");
puts("2. Trier la pile");
puts("3. Supprimer une valeur");
puts("4. Supprimer les doublants");
puts("5. Afficher le nombre d'occurrence d'une valeur");
puts("6. Afficher la pile.");
puts("7. Quitter.");
puts(" #####");
printf("\t Votre choix : ");scanf("%c",&c);
switch(c){
case '1': system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '2': system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '3': system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '4': system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '5':system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '6':system("cls");puts("\n\t to be completed !!!");
getche(); break;
case '7': printf("Voulez vous quitter le programme o/n
??\n");
c =(char)getch(); break;
}
}
system("cls");
}
return 0;
}
```