

Atelier 4: Les Piles et les Files en C

Compétences à atteindre

- C1.** Savoir déclarer, initialiser, empiler, dépiler, afficher et vider une pile
- C2.** Savoir déclarer, initialiser, enfiler, défiler, afficher et vider une file
- C3.** Faire la différence entre les structures étudiées (Listes chaînées, piles et files) et les exploiter dans la situation convenable

Exercice n°1 : File d'étudiants

Soit à manipuler la structure file. Cette liste est composée par un ensemble de cellules contenant les informations des étudiants qui sont chaînées par un pointeur (suivant). Chaque étudiant est caractérisé par son numéro d'inscription (entier) son nom (chaîne de caractères), son prénom (chaîne de caractères) et sa moyenne (réel).

1. Définir la structure étudiant et la structure File.
2. Ecrire une fonction d'initialisation de la file pointée par un pointeur F. F est l'adresse d'une structure de type FILE: **void initialiser (File * F).**
3. Ecrire une fonction qui teste et retourne si la file F est vide ou pas :
int fileVide (File F).
4. Écrire la fonction **enfiler_etudiant** qui ajoute un étudiant à la file :
void enfiler_etudiant (File * F, Etudiant etud).
5. Écrire la fonction **Affiche_etudiants** qui affiche tous les étudiants de la file :
void Affiche_etudiants (File F).
6. Écrire la fonction **defiler_etudiant** qui défile (supprime) un étudiant
void defiler_etudiant (File * F).
7. Écrire un programme principal permettant de:
 - Saisir le nombre d'étudiants de la file.
 - Créer la file des étudiants.
 - Enfiler n étudiants dans la file.
 - Afficher tous les étudiants de la file.
 - défiler p (p<n) étudiants de la file

Exercice n°2 : Pile de produits

Un magasin a décidé d'informatiser sa gestion de stock, pour ce faire, on a choisi d'utiliser une pile qui contient l'ensemble des produits commercialisés. Chaque produit sera caractérisé par son code et son libellé.

1. Ecrire la déclaration de la **Pile** de produits.
2. Ecrire une fonction **void pile_push(Pile *PE, Produit P)** qui empile un produit en tête de pile.
3. Ecrire une fonction **void affiche_produits(Pile PE)** qui affiche les produits de la pile
4. Ecrire une fonction **void pile_pop(Pile *PE, Produit * P)** qui dépile un élément de la tête de pile
8. Écrire un programme principal permettant de:
 - Saisir le nombre de produits de la pile.
 - Créer une pile vide.
 - Empiler n produits dans la pile.
 - Afficher tous les produits.
 - Dépiler p ($p < n$) produits de la pile.

Travail à rendre

L'objectif de l'exercice est de reconnaître si une expression est bien parenthésée avec deux types de parenthèses '[' et '('.

1. Proposer une structure à utiliser pour résoudre le problème.
2. Ecrire la fonction **int bienParenthésée(char * expr)** qui retourne 1 si l'expression saisie en paramètre est bien parenthésée ou 0 sinon.