

TP 7: Les files

Exercice 1: File statique

On souhaite gérer une file d'étudiants à l'aide d'un *tableau*.

1- Créer une 1^{ère} *structure Etudiant* qui contient les champs: *cne* (entier), un *nom* et un *prenom* (pointeur sur un caractère) et une 2^{ème} *structure File* avec les champs *tete* et *queue* (entiers) et un tableau d'étudiants *tab* de taille *dimension*.

Ajouter les fonctions suivantes pour gérer la file:

2- *void init_file (File *F)* qui initialise la *tete*, la *queue* et la *taille* de la file à 0.

3- *int file_vide (File *F)* qui teste si la file est vide.

4- *int file_pleine (File *F)* qui teste si la file est pleine.

5- *int enfileur (File *F, int nouveau_cne, char * nouveau_nom, char* nouveau_prenom)* qui ajoute un nouveau étudiant (*nouveau_cne*, *nouveau_nom* et *nouveau_prenom*) dans la file. Utiliser les fonctions *file_vide* et *file_pleine*. Cette fonction retourne 1 en cas d'enfilement réussi, -1 en cas de problème dans la réservation de la mémoire avec *malloc* et 0 si la file est saturée.

6- *void remplir (File *F)* qui permet de remplir la file par *n* étudiants. Utiliser la fonction *enfileur* précédente pour enfileur plusieurs étudiants.

7- *int defiler (File *F)* qui retire le premier étudiant inséré dans la file. Cette fonction retourne le *cne* de l'étudiant retiré en cas de défilement réussi et -1 sinon.

8- *void affichage(File *F)* qui affiche le contenu de la file.

9- *int rechercher (File * F, int cne)* qui recherche un étudiant dans la file. La recherche se fait par *cne*. La fonction retourne sa position dans la file s'il existe et -1 sinon.

10- *void menu(File *F)* qui contient un menu textuel qui permet à l'utilisateur de choisir l'opération à réaliser: 1) initialiser la file, 2) remplir la file, 3) défiler, 4) afficher la file, 5) rechercher dans la file.

11- Ajouter le programme principal qui fait appel au menu.

Exercice 2: File dynamique

Ecrire un programme qui gère une file d'étudiants à l'aide d'une liste chaînée. Pour cela, créer une 1^{ère} *structure Etudiant* qui contient les champs: *cne* (entier), un *nom* et un *prenom* (pointeur sur un caractère) et une variable *suivant* de type *pointeur* sur la structure *Etudiant*. Et créer une 2^{ème} *structure File* avec des champs *tete* et *queue* de type *pointeurs* sur la structure *Etudiant* et un entier *taille*.

La file est gérée à l'aide des fonctions suivantes :

1- *void initialiser(File * F)* qui initialise les pointeurs *tete* et *queue* à NULL et *taille* à 0.

2- *int enfileur (File *F, int nouveau_cne, char *nouveau_nom, char *nouveau_prenom)* qui ajoute un nouveau étudiant (*nouveau_cne*, *nouveau_nom* et *nouveau_prenom*) à la file, modifie le pointeur *queue* et incrémente la *taille* de la file. La fonction retourne -1 si l'enfilement a échoué et 0 sinon.

3- *int defiler (File *F)* qui retire le 1^{er} élément inséré dans la file, modifie le pointeur *tete* et décrémente la *taille* de la file. La fonction retourne -1 si le défilement a échoué et 0 sinon.

4- *void afficher_recu(File *F, Etudiant *courant, int taille, int i)* qui affiche les données de la file de manière *réursive*.

5- Ecrire le programme principal qui fait appel aux fonctions précédentes. Ajouter un menu.