On s'intéresse à la gestion d'impression des travaux d'un groupe d'étudiants. Ils sont munis d'une priorité dépendant des utilisateurs. Cette priorité est exprimée sous forme d'un entier de 1 pour les plus prioritaires à 5 pour les moins prioritaires. Les travaux sont placés dans <u>UNE LISTE CIRCULAIRE</u> <u>AVEC UNE TETE MOBILE</u>. Un élément de cette liste est de cette forme : nom_etudiant[30], nom_fichier[50] et priorité.

- 1. Donnez la définition (le type) de cette structure.
- 2. Ecrire une fonction qui construit la liste circulaire pour les données suivante :

```
char Nom_Etudiant[5][30]={"ALI", "MIRA", "SAMY", "KAMEL", "ZAHIYA"};
char Nom_Fichier[5][30]={"algo.pdf", "BD.doc", "archi.ps", "logique.txt", "proba.zip"};
int priorite[5]= {1, 3, 4, 2, 2};
```

REMARQUE : on transmet à cette fonction : <u>un seul</u> nom d'étudiant, <u>un seul</u> fichier à imprimer et <u>une seule</u> priorité.

3. Ecrire une fonction qui affiche cette liste

int main()

4. Ecrire la fonction cherche_Min_Adr_Prec() qui rend l'adresse de l'étudiant le plus prioritaire (adrMin) et déplace la queue de la liste sur l'élément précédent cet étudiant. L'entête de cette fonction est donné comme suit :

```
ListeC cherche Min Adr Prec(ListeC queue, ListeC *adrMin)
```

5. Ecrire la fonction supprim_Apres() qui supprime un élément se trouvant à l'adresse P. L'entête de cette fonction est donné comme suit:

```
void supprim Apres(ListeC * prd, ListeC P)
```

6. Ecrire le programme C en suivant les commentaires suivants :

```
/*******DECLARATION DES VARIABLES ********
/*******REMPLISSAGE DE LA LISTE CIRCULAIRE*******
/*******AFFICHAGE DE LA LISTE CIRCULAIRE*******

Tant que (Condition(s))
Faire
/********CHERCHER LE MINIMUM********
//
/*******SUPPRIMER LE MINIMUM******
//
/*******AFFICHAGE DE LA LISTE CIRCULAIRE******
//
Fait
/********
return 0;
```

Exercice 2: Etant donné une matrice carrée T[N][N] où N > 1 et N est impair. On veut représenter un triangle dans la moitié supérieure de cette matrice en mettant à jour certaines cases de la matrice T. Pour effectuer le remplissage du triangle on procède comme suit: à chaque fois qu'on met à jour une case par un '1', on sauvegarde les coordonnées entourant cette case (gauche, droite, haut et bas) dans une pile.

- 1. Définissez le type « coordo » : les coordonnées (x, y) d'une matrice et donnez le type de la pile.
- 2. Réécrivez les deux actions Désempiler et Empiler en tenant compte du nouveau type « coordo ».
- 3. Ecrivez une action qui place des '1' dans la partie supérieure de la matrice T en suivant le principe énoncé ci-dessus.

<u>Note</u>: on prend toujours comme configuration initiale les deux côtés du triangle et une case de départ au milieu du triangle exemple: T [2][4] \neq 1 (voir par exemple la figure 1). Le résultat final est donné par la figure 2.



