

CORRIGE TP N° : 3

LES FILES

ENSEIGNANTE : MME ONS BEN ROMDHANE
MATIERE : ALGORITHMIQUE AVANCE

SEMESTRE : 2- 2023/2024
GROUPES : 3^{EME} INFO H, I & J

```
#include <stdio.h>
#include <stdlib.h>

struct cellule
{   int elem;
    struct cellule * succ;
};

struct file
{   cellule *tete;
    cellule *queue;
};

//Prototypes des primitives fondamentales
void init (file * f);
bool  vide (file f);
void PROC_enfiler (file * f, int e);
file FN_enfiler (file f, int e);
void afficher (file f);
void PROC_defiler (file *f, int *e);
void PROC_defiler1 (file *f, int **e);

int main()
{   file f;
    //Initialiser la file f à vide
    init(&f);
```

```

// Création de la file (10,20,30). 10 est la tête et 30
//est la queue.
PROC_enfiler (&f, 10); //ou bien f=FN_enfiler(f, 10);
PROC_enfiler (&f, 20);
PROC_enfiler (&f, 30)

//Afficher les éléments de la file
printf("\nLa file est :");
afficher(f);

//Défiler la file (version 1: File non vide)
int n;
if (f.tete != NULL) // La file n'est pas vide
{
    PROC_defiler(&f, &n);
    printf("\nLe premier sorti de la file est : %d",n);
    //Affichage de la file après la sortie du premier
    printf("\nLa file sera :");
    afficher(f);
}

//Défiler la file (version 2: File vide ou non vide)
int *sf;
PROC_defiler1(&f, &sf);
if (sf!=NULL)
{
    printf("\nLe premier sorti de la file est : %d", *sf);
    //Affichage de la file après la sortie du premier
    printf("\n\nLa file sera : ");

    afficher(f);
}
else printf("Liste vide");
}

```

```

void init (file * f)
{
    f->tete = NULL;
    f->queue = NULL;
}

bool vide(file f)
{
    return (f.tete==NULL);
}

void PROC_enfiler (file * f, int e)
{ /* Postcond : f' ayant tous les éléments de la file f et
   en queue un élément contenant e.
   Si f contient(a1, a2, ... an), elle devient(a1, a2, ... an, e).
   Premier(f') = a1. Le paramètre f est passé par variable car
   la file peut être vide.*/

    cellule * cel;
    cel=(cellule*) malloc (sizeof(cellule));
    cel->elem = e;
    cel->succ = NULL;
    if (vide(*f))
        f->tete = cel;
    else
        f->queue->succ = cel;

    f->queue = cel;
}

file FN_enfiler (file f, int e)
{ /* Postcond : Cette fonction ajoute le nouvel élément e
   à la fin (queue) de f et retourne la nouvelle file f. */
    cellule *cel;
    cel=(cellule*) malloc (sizeof(cellule));

```

```

    cel->elem = e;
    cel->succ = NULL;
    if (vide(f))
        f.tete = cel;
    else
        f.queue->succ = cel; //Fin if

    f.queue = cel;

    return f;
}

void afficher (file f)
{ // Afficher tous les éléments de la file à partir de la
  tête

    if (vide(f))
        printf("File Vide\n");
    else
        while(!vide(f))
        {   printf("%d \t", f.tete->elem);
            f.tete = f.tete->succ;
        }
}

void PROC_defiler (file *f, int *e)
{ /* Précond      : Non Vide (f)
   Postcond : f' ayant tous les éléments de la file f
               sans le premier élément.
   Si f contient(a1, a2, ... an), elle devient(a2, ... an)
   et e = a1 */
    cellule *cel;
    cel = f->tete;
    *e = cel->elem;
    f->tete = cel->succ;
}

```

```

    if (f->tete==NULL)
        f->queue = NULL;
    free(cel);
}

void PROC_defiler1 (file *f, int* *e)
{ /* Précond : Aucune.
   Postcond : Supprimer la tête de la file p, e est un
               pointeur sur l'entier récupéré. Si la file
               est vide, e reçoit NULL. */

    cellule *cel;
    if (vide(*f))
        e=NULL;
    else
    { *e=(int*)malloc(sizeof(int));
      cel = f->tete;
      **e = cel->elem;
      f->tete = cel->succ;
      if (f->tete==NULL)
          f->queue = NULL;
      free(cel);
    }
}

```