

CORRIGE TP N° : 2
LISTES SIMPLEMENT CHAINEES DE FILMS

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>
#include <string.h>
struct film
{   char titre[20] ;
    int cout ;      // Coût du film exprimé en milliers de dinars
} ;

struct cellule
{   film info;
    struct cellule * suiv;
} ;

typedef struct cellule * liste ;

int menu()
{   int c;
    printf(" \n \n 1 Saisie des films\n");
    printf(" 2 Afficher tous les films\n");
    printf(" 3 Suppression d'un film\n");
    printf(" 4 Quitter le programme\n");

    printf(" \nDonner votre choix ");
    scanf("%d",&c);
    return c;
}
```

```

liste recherche_film (liste tete,char tfilm [])
{
// Version 1
while (tete)
{
    if (strcmp(tete->info.titre,tfilm)==0)
        return tete;
    tete=tete->suiv;
}
return NULL;
}
// Version 2
while (tete && strcmp(tete->info.titre,tfilm)!=0)
    tete=tete->suiv;

return tete;

// Version 3
liste pt=tete;  int trouve=0;
while (pt && trouve ==0)
{
    if (strcmp(pt->info.titre,tfilm)==0)
        trouve=1;
    else pt=pt->suiv;
}

return pt;

// Version 4 recursive
if (!tete)
    return NULL;
else
    if (strcmp(tete->info.titre,tfilm)==0)
        return tete;

```

```

        else
            return recherche_film (tete->suiv,tfilm);
    }

```

liste inserer_debut (liste tete)

```

{ liste nouv, pt;

```

//Allocation dynamique de la nouvelle cellule

```

nouv=( liste) malloc (sizeof(cellule));

```

```

if (nouv==NULL)

```

```

    { printf("pas d'allocation de memoire");

```

```

        exit(-1);

```

```

    }

```

```

else

```

```

{ //Remplissage de la nouvelle cellule

```

```

    do

```

```

    {printf("\n Titre du film : ");

```

```

        scanf("%20s",nouv->info.titre);

```

```

        pt=recherche_film(tete,nouv->info.titre);

```

```

        if (pt)

```

```

            printf("\n Ce film existe deja");

```

```

        }while(pt);

```

```

        printf("\n Cout du film : ");

```

```

        scanf("%d",&nouv->info.cout);

```

```

    /*Insertion de la nouvelle cellule dans la liste
    et maj des pointeurs*/
    nouv->suiv=tete;
    tete=nouv;
}
return tete;
}

```

```

liste saisie_films (liste tete)
{
    int nb,i;

    printf("\n nombre de films : ");
    scanf("%d",&nb);

    for(i=0;i<nb;i++)
        tete = inserer_debut (tete);

    return tete;
}

```

```

void affiche(liste tete)
{
    if (tete==NULL)
        printf("\nliste vide");
    else
        {
            while (tete!=NULL)
            {
                printf("\ntitre : %s ",tete->info.titre);
                printf("cout : %d  \n\n",tete->info.cout);
                tete=tete->suiv;
            }
        }
}

```

```

liste supprimer_debut(liste tete)
{
    if (tete)
    {
        liste pt=tete;
        tete=tete->suiv;
        free(pt);
    }
    return tete;
}

```

```

liste supprimer_fin(liste tete)
{
    liste pt=tete;
    if (tete)
    {
        if(!tete->suiv )
            tete=supprimer_debut(tete);
        else
        {
            while(pt->suiv->suiv)
                pt=pt->suiv;

            free(pt->suiv);
            pt->suiv=NULL;
        }
    }
    return tete;
}

```

```

void supprimer_milieu(liste tete, liste pt)
{

/*pt est l'adresse de la cellule à supprimer
  La suppression n'est ni au début ni à la fin
  la liste est suppose non vide */

liste m=tete;

while(m->suiv!=pt)
    m=m->suiv;

m->suiv=pt->suiv;
free(pt);
}

liste supprimer_film(liste tete, char titre[20])
{
    liste pt;

    pt=recherche_film(tete, titre);

    if(pt==NULL)
        printf(" \n Film n'existe pas");
    else
    {
        if (pt==tete)
            tete=supprimer_debut(tete);
        else
        {
            if(pt->suiv==NULL)
                tete=supprimer_fin(tete);
            else
                supprimer_milieu(tete, pt);
        }
    }
}

```

```

        printf("\n Le film est supprime\n");
    }
    return tete;
}

/* Version 2 de supprimer_film sans appeler
les modules supprimer_debut, supprimer_milieu et
supprimer_fin */

liste supprimer_film2(liste tete, char titre[])
{ /*Parcourir la liste jusqu'à soit trouver la cellule
à supprimer soit arriver à la fin sans la trouver*/

    liste p=tete;
    liste prec = NULL;

    // Si la cellule à supprimer est en tête
    if (tete != NULL && strcmp(tete->info.titre ,titre)==0)
    {
        tete = p->suiv;
        free(p);
        printf("Le film est supprimé.\n");
    }
    else
    {
        // Recherche de la cellule à supprimer
        while (p != NULL && strcmp(p->info.titre ,titre)!=0)
        {
            prec = p;
            p = p->suiv;
        }
    }
}

```

```

// Si le film n'existe pas dans la liste
if (p == NULL)
    printf("Le film n'existe pas.\n");

else
{
    /* Liaison de la cellule précédente avec la
    cellule suivante de celle à supprimer*/
    prec->suiv = p->suiv;

    // Libération de la mémoire de la cellule à
    supprimer
    free(p);
    printf("Le film est supprimé de la liste.\n");
}
return tete;
}
}

int main()
{ int c; liste tete=NULL;
  char titre[20];

  system("cls");
  do
  { c=menu();
    switch(c)
    {case 1 : tete=saisie_films(tete); break;
     case 2 : affiche(tete); break;
     case 3 : printf("\nDonner le titre du film à
                     supprimer: ");

```



```
scanf("%s",titre);

tete=supprimer_film(tete,titre);
break;

case 4 : printf("Fin du travail"); break;
default : printf("choix errone"); break;
}
}while (c!=4);
}
```