

TP Allocation dynamique

Toutes les fonctions doivent être testées dans un main

une matrice est définie par la structure suivante :

```
struct matrice  
{  
    int nbl ;  
    int nbc ;  
    double *T ;  
  
};
```

Le tableau T doit contenir ***nbl*** x ***nbc*** cases de type double. Pour accéder à la case de la ligne ***i*** et la colonne ***j*** on fait ***T[i*nbc+j]***.

1- a Écrire une fonction qui prend en argument n et m (nombre de lignes et nombre de colonnes) et qui renvoie un pointeur sur une matrice :

matrice *construit_matrice(int n, int m) ;

1-b Ecrire la version suivante :

void construit_matrice(matrice &A, int n, int m) ;

2- Ecrire une fonction qui transpose une matrice :

void transpose(matrice &A) ;

Cette fonction modifie la matrice A elle même.

3-

a- Écrire une fonction qui prend en argument deux matrices A et B et qui renvoie un pointeur sur une nouvelle matrice contenant la somme des deux matrices :

matrice *add(matrice &A, matrice &B) ;

b- Écrire une autre version de ***add***, ***add_pointeur*** :

matrice *add_pointeur(matrice *A, matrice *B) ;

Dans tous les cas si l'opération n'est pas possible affichez un message et arrêtez le programme.

4- Écrire une fonction qui prend en argument deux matrices A et B et qui renvoie un pointeur sur une nouvelle matrice contenant le produit des deux matrices :

matrice *mult(matrice &A, matrice &B) ;

Si l'opération n'est pas possible affichez un message et arrêtez le programme.

5- a- Écrire une fonction qui libère une matrice : ***void Free_matrice(matrice *A) ;***

5- b- Écrire une fonction qui libère une matrice : ***void Free_matrice(matrice &A) ;***