COMPTE RENDU EXO 3 TP 1 RN40

BARTHELME ALEXANDRE

Mon code complet est disponible sur mon GitHub à l'adresse :
https://github.com/Iron68A/RN40/tree/main/TP

Ou en Annexe par capture d'écran

I) Les matrices

Les matrices sont définies comme des tableaux à deux dimensions. Les matrices sont forcément carrées ici.

Les matrices sont définies comme contenant que des entiers, par facilités et pour limiter les problèmes. Elles sont définies comme : int matrice[n][n], avec n la longueur de la ligne et de la colonne.

Ce « n » est demander à l'utilisateur dès le début du programme, une fois qu'il a choisi son opération voulue. Pour remplir les matrices, l'utilisateur entre les valeurs. Il faut parcourir les deux tableaux

(toute la ligne, puis toute la colonne) pour remplir la matrice. C'est une double boucle « for ».

II) Les calculs

Les différentes fonctions prennent en arguments les matrices nécessaires et une matrice « resultat » pour pouvoir afficher le résultat dans le « main » (sinon le calcul est fait mais rien n'est renvoyer à l'utilisateur)

Les fonctions prennent aussi en arguments l'entier n correspondant à la taille des matrices pour les calculs. Celle de division et de multiplication ont en plus le nombre donner par l'utilisateur.

D'ailleurs, si l'utilisateur choisi division ou multiplication, le programme ne lui demande qu'une matrice. Les choix sont traités par un « switch » plutôt que les « if » car on peut utiliser la valeur saisie par l'utilisateur, et qu'il y a que 4 choix. Les calculs eux-mêmes consistent à prendre chacun des éléments des tableaux (ligne et colonne) et de faire l'opération voulue avec l'autre matrice ou un nombre.

III) Résultat

Le résultat est affiché sous forme de matrice après l'opération :

```
PS D:\GitHub\RN40\TP> cd "d:\GitHub\RN40\TP\" ; if ($?) { gcc matrice.c -o matrice } ; if ($?) { .\matrice }
quel operation voulez vous faire ?
 1: addition
 2: soustraction
 3: mutliplication par un nombre
 4: division par un nombre
 1
Donner la taille des matrices: 2
Donner les elements de la matrice 1:
donner l'element 0 0: 2
donner l'element 0 1: 2
donner l'element 1 0: 2
donner l'element 1 1: 2
Donner les elements de la matrice 2:
Donner l'element 0 0: 3
Donner l'element 0 1: 3
Donner l'element 1 0: 3
Donner l'element 1 1: 3
La matrice resultat est:
5 5
5 5
PS D:\GitHub\RN40\TP>
```

Pour toutes questions ou informations complémentaires, me contacter par mail :

alexandre.barthelme@utbm.fr

ANNEXE

```c
#include<stdio.h>
#include<math.h>
/*CODE PRIVE
  Programmer en C
  TP 1 RN40
  Sujet dans le repo github

BARTHELME ALEXANDRE

*/

//declaration des fonctions avant le main, car fonctions apres ( par habitude )
void addition(int n, int matrice1[n][n], int matrice2[n][n], int resultat[n][n]);
void soustraction(int n, int matrice1[n][n], int matrice2[n][n], int resultat[n][n]);
void multiparnbr(int n, int matrice[n][n], int nbr, int resultat[n][n]);
void divparnbr(int n, int matrice[n][n], int nbr, int resultat[n][n]);

int main(){
    int n,i,j;
    int selec;
    int scal;

    //choix
    printf("quel operation voulez vous faire ? \n 1: addition\n 2: soustraction\n 3: mutliplication par un nombre\n 4: division par un nombre \n ");
    scanf("%d",&selec);


    //taille
    printf("Donner la taille des matrices: ");
    scanf("%d",&n);
    int matrice1[n][n],matrice2[n][n],resultat[n][n];

    //remplissage des matrices
    printf("Donner les elements de la matrice 1:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("donner l'element %d %d: ",i,j);
            scanf("%d",&matrice1[i][j]);
        }
    }
    if (selec!=3 && selec!=4){
    printf("Donner les elements de la matrice 2:\n");
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            printf("Donner l'element %d %d: ",i,j);
            scanf("%d",&matrice2[i][j]);
        }
    }
    }
```

```c
    //calcul en fct du choix
    switch (selec){
      case 1:
        addition(n,matrice1,matrice2,resultat);
        break;
      case 2:
        soustraction(n,matrice1,matrice2,resultat);
        break;
      case 3:
        printf("Donner le nombre par lequel vous voulez multiplier la matrice: ");
        scanf("%d",&scal);
        multiparnbr(n,matrice1,scal,resultat);
        break;
      case 4:
        printf("Donner le nombre par lequel vous voulez diviser la matrice: ");
        scanf("%d",&scal);
        divparnbr(n,matrice1,scal,resultat);
        break;
      default:
        printf("Un probleme est survenu, relancer le programme en suivant les instructions");
    }
```

```c
//FONCTIONS DE CALCULS


void addition(int n, int matrice1[n][n], int matrice2[n][n], int resultat[n][n]){
    int i,j;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            resultat[i][j]=matrice1[i][j]+matrice2[i][j];
        }
    }
}

void soustraction(int n, int matrice1[n][n], int matrice2[n][n], int resultat[n][n]){
    int i,j;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            resultat[i][j]=matrice1[i][j]-matrice2[i][j];
        }
    }
}
```

```c
void multiparnbr(int n, int matrice[n][n], int nbr, int resultat[n][n]){
    int i,j;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            resultat[i][j]=matrice[i][j]*nbr;
        }
    }
}

void divparnbr(int n, int matrice[n][n], int nbr, int resultat[n][n]){
    int i,j;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            resultat[i][j]=matrice[i][j]/nbr;
        }
    }
}
```