//Realizzare le operazioni di somma di 2 matrici e prodotto scalare di una matrice per un numero reale.

//Il tipo di dato matrice deve essere rappresentato come un vettore di reali a 2 dimensioni variabili.

#include<stdio.h>

#include<stdlib.h>

void CaricareMatrice(int x, int y, int M[x][y]);//Funzione che carica la matrice

void SommareMatrici(int x, int y, int M1[x][y], int M2[x][y]);/\*Funzione che crea una matrice delle stesse dimensione delle altre due e inserisce

in ogni posizione la somma degli elementi delle due matrici\*/

void MoltiplicareMatricePerN(int p, int x, int y, int M[x][y]);//Funzione che moltiplica ogni elemento di una matrice scelta dall'utente per un numero scelto dall'utente

void VisualizzareMatrice(int x, int y, int M[x][y]);

int main()

{

int n1=0,m1=1,n2=1,m2=0,moltiplicatore=0,scelta=4;/\*n1=numero righe della prima matrice

m1=numero colonne della prima matrice

n2=numero righe della seconda matrice

m2=numero colonne della seconda matrice

moltiplicatore=numero scelto dall'utente che andrà a moltiplicare una matrice

scelta=variabile che servirà a contenere quale matrice andrà a moltiplicare

(scelta=4 valore preimpostato)

\*/

printf("\n Inserire le dimensioni delle due matrici (le dimensioni delle righe e delle colonne delle due matrici devono essere uguali) \n");

while((n1!=n2)&&(m1!=m2))

{

printf("\n Inserire numero righe prima matrice \n");

scanf("%d",&n1);

printf("\n Inserire numero colonne prima matrice \n");

scanf("%d",&m1);

printf("\n Inserire numero righe seconda matrice \n");

scanf("%d",&n2);

printf("\n Inserire numero colonne seconda matrice \n");

scanf("%d",&m2);

}

int Mat1[n1][m1];

int Mat2[n2][m2];

CaricareMatrice(n1,m1,Mat1);

VisualizzareMatrice(n1,m1,Mat1);

CaricareMatrice(n2,m2,Mat2);

VisualizzareMatrice(n2,m2,Mat2);

printf("\n SOMMA DELLE MATRICI \n");

SommareMatrici(n1,m1,Mat1,Mat2);

printf("\n Inserire il numero che dovra' moltiplicare la matrice: \n");

scanf("%d",&moltiplicatore);

printf("\n Inserire 1 per moltiplicare la prima matrice, 2 per moltiplicare la seconda matrice \n");

scanf("%d",&scelta);

if(scelta==1)

MoltiplicareMatricePerN(moltiplicatore,n1,m1,Mat1);

else

MoltiplicareMatricePerN(moltiplicatore,n2,m2,Mat2);

}

void CaricareMatrice(int x, int y, int M[x][y])

{

int i,j;

i=0;

printf("\nInserimento matrice %dx%d\n",x,y);

while(i<x)

{

j=0;

printf("\n");

while(j<y)

{

printf("\nMat[%d][%d]: ",i,j);

scanf("%d",&M[i][j]);

j++;

}

i++;

}

}

void VisualizzareMatrice(int x, int y, int M[x][y])

{

int i,j;

i=0;

printf("\nVisualizzazione Matrice:");

while(i<x){

j=0;

printf("\n");

while(j<y){

printf("%d\t",M[i][j]);

j++;

}

i++;

}

}

void SommareMatrici(int x, int y, int M1[x][y], int M2[x][y])

{

int i,j,l,m;

int M3[x][y];

i=0;

while(i<x)

{

j=0;

while(j<y)

{

M3[i][j]=(M1[i][j])+(M2[i][j]);

j++;

}

i++;

}

VisualizzareMatrice(x,y,M3);

}

void MoltiplicareMatricePerN(int p, int x, int y, int M[x][y])

{

int i,j;

i=0;

while(i<x)

{

j=0;

while(j<y)

{

M[i][j]=M[i][j]\*p;

j++;

}

i++;

}

VisualizzareMatrice(x,y,M);

}

**Funziona**