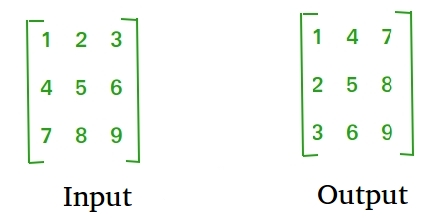
Write a menu driven program to perform following matrix operations

1. Transpose

Transpose of a matrix is the change of rows to columns and columns to rows. Here we have a matrix a[][], which we change from a[i][j] to a[j][i]

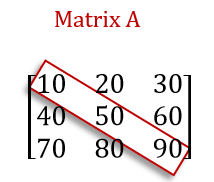


1. Multiplication

In matrix multiplication, we are asking number of rows and columns for the first matrix and the second matrix then we enter the elements of the first and second matrices. Remember that matrix multiplication here we have the same number of rows and columns.

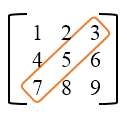
1. Addition of Major Diagonal Elements

In this program, we add the elements from the top left corner to the bottom right corner. (L to R). Logic is if(i==j) then sum=sum+mat[i][j]. Then you print ‘sum’.



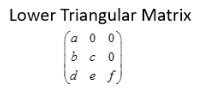
1. Addition of Minor Diagonal Elements

In this program, we add the elements from the top right corner to the bottom left corner. (R to L). Logic is if(r+c==2) then add=add+matrix[r][c]. Then you print ‘add’.



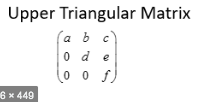
1. Lower Triangular Matrix

Enter number of rows and columns then enter elements then print upper triangular matrix with two for loops where If (i<j) then print zero or else print(matrix)



1. Upper Triangular Matrix

Enter number of rows and columns then enter elements then print upper triangular matrix with two for loops where If (i>j) then print zero or else print(matrix)



1. Find whether Matrix is Symmetric or Not?

Ask for number of rows and columns, then enter elements of 2d array then print the actual array and then transpose that array using a[i][j]=a[j][i]. Plus, all of that should be under two for loops. In the end use this condition under two for loops if(a[i][j]!=a[j][i]) then print matrix is not symmetric. Then in the end type print matrix is symmetric.

#include<stdio.h>

int main()

{

int a,b,d,e,mat[10][10];

int s,t,u,v,a3[10][10],b3[10][10],mul[10][10];

int x,y,z,w,mat1[10][10],sum1 = 0;

int g,h,k,l,mat2[10][10],add = 0;

int row,col,m3,n3,mat3[10][10];

int rows,cols,m4,n4,mat4[10][10];

int rr,cc,f,o,sym,a4[10][10],b4[10][10];

while(1)

{

printf("\n1. Matrix Tranpose\n");

printf("\n2. Matrix Multiplication\n");

printf("\n3. Addition of Major Diagonal Elements\n");

printf("\n4. Addition of Minor Diagonal Elements\n");

printf("\n5. Upper Triangular Matrix\n");

printf("\n6. Lower Triangular Matrix\n");

printf("\n7. Matrix is symmetric or not\n");

printf("\nEnter a choice:\n");

scanf("%d",&e);

switch(e)

{

case 1:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&d,&e);

printf("Enter elements of Matrix:\n");

for(a=0;a<d;a++)

{

for(b=0;b<e;b++)

{

scanf("%d",&mat[a][b]);

}

}

printf("\nBefore transposing:\n");

for(a=0;a<d;a++)

{

for(b=0;b<e;b++)

{

printf("%d\t",mat[a][b]);

}

printf("\n");

}

printf("\nAfter transposing:\n");

for(a=0;a<d;a++)

{

for(b=0;b<e;b++)

{

printf("%d\t",mat[b][a]);

}

printf("\n");

}

break;

case 2:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&u,&v);

printf("Enter elements of Matrix A:\n");

for(s=0;s<u;s++)

{

for(t=0;t<v;t++)

{

scanf("%d",&a3[s][t]);

}

}

printf("Enter elements of Matrix B:\n");

for(s=0;s<u;s++)

{

for(t=0;t<v;t++)

{

scanf("%d",&b3[s][t]);

}

}

for(s=0;s<u;s++)

{

for(t=0;t<v;t++)

{

mul[s][t]=a3[s][t]\*b3[s][t];

printf("%d\t",mul[s][t]);

}

printf("\n");

}

break;

case 3:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&w,&v);

printf("Enter elements of matrix:\n");

for(x = 0; x < w; x++)

{

for(y = 0; y < v; y++)

{

scanf("%d", &mat[x][y]);

}

}

printf("\nActual matrix\n");

for(x = 0; x < w; x++)

{

for(y = 0; y < v; y++)

{

printf("%d\t", mat[x][y]);

}

printf("\n");

}

for(x = 0; x < w; x++)

{

for(y = 0; y < v; y++)

{

if(x == y)

sum1 = sum1 + mat[x][y];

}

}

printf("\n The sum of Major Diagonal Elements is \t%d", sum1);

printf("\n");

break;

case 4:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&g,&h);

printf("Enter elements of matrix:\n");

for(k = 0; k < g; k++)

{

for(l = 0;l < h; l++)

{

scanf("%d", &mat[k][l]);

}

}

printf("\nActual matrix\n");

for(k = 0; k < g; k++)

{

for(l = 0; l < h; l++)

{

printf("%d\t", mat[k][l]);

}

printf("\n");

}

for(k = 0; k < g; k++)

{

for(l = 0; l < h; l++)

{

if(k + l == 2)

add = add + mat[k][l];

}

}

printf("\n The sum of Minor Diagonal Elements is \t%d", add);

printf("\n");

break;

case 5:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&m3,&n3);

printf("Enter elements of Matrix:\n");

for(row=0;row<m3;row++)

{

for(col=0;col<n3;col++)

{

scanf("%d",&mat3[row][col]);

}

}

printf("The Upper Triangular Matrix:\n");

for(row=0;row<m3;row++)

{

for(col=0;col<n3;col++)

{

if(row>col)

{

printf("0");

}

else

{

printf("%d",mat3[row][col]);

}

}

printf("\n");

}

break;

case 6:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&m4,&n4);

printf("Enter elements of Matrix:\n");

for(rows=0;rows<m4;rows++)

{

for(cols=0;cols<n4;cols++)

{

scanf("%d",&mat4[rows][cols]);

}

}

printf("The Lower Triangular Matrix:\n");

for(rows=0;rows<m4;rows++)

{

for(cols=0;cols<n4;cols++)

{

if(rows<cols)

{

printf("0");

}

else

{

printf("%d",mat[rows][cols]);

}

}

printf("\n");

}

break;

case 7:printf("\nEnter number of rows and columns:\n");

scanf("%d%d",&f,&o);

printf("Enter elements of matrix:\n");

for(rr=0;rr<f;rr++)

{

for(cc=0;cc<o;cc++)

{

scanf("%d",&a4[rr][cc]);

}

}

printf("Original Matrix:\n");

for(rr=0;rr<f;rr++)

{

for(cc=0;cc<o;cc++)

{

printf("%d\t",a4[rr][cc]);

}

printf("\n");

}

printf("Tranpose of matrix:\n");

for(rr=0;rr<f;rr++)

{

for(cc=0;cc<o;cc++)

{

b4[rr][cc]=a4[cc][rr];

printf("%d\t",b4[rr][cc]);

}

printf("\n");

}

for(rr=0;rr<f;rr++)

{

for(cc=0;cc<o;cc++)

{

if(a4[rr][cc]!=b4[rr][cc]);

{

printf("Matrix is not symmetric");

return 0;

}

}

}

printf("Matrix is symmetric");

break;

default:printf("Enter choice again:");

}

}

}