**25 April Home questions**

#1. WAP to check whether a given matrix is symmetric or not.

Code:

#include <stdio.h>

int main()

{

    int i\_285,j\_285,m\_285,n\_285,A\_285[10][10],B\_285[10][10],count\_285;

    printf("Provide the order of your square matrix=>");

    scanf("%d%d",&m\_285,&n\_285);

    printf("Give the elements of your square matrix=>\n");

    //taking a fresh input

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            printf("A[%d][%d]=",i\_285,j\_285);

            scanf("%d",&A\_285[i\_285][j\_285]);

        }

    }

    //Storing the transpose of the input matrix in B

    for(j\_285=0;j\_285<n\_285;j\_285++)

    {

        for(i\_285=0;i\_285<m\_285;i\_285++)

        {

            B\_285[i\_285][j\_285]=A\_285[j\_285][i\_285];

        }

    }

    //printing the matrix

    printf("Your matrix is\n");

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            printf("%d\t",A\_285[i\_285][j\_285]);

            if(j\_285==(n\_285-1))

            printf("\n");

        }

    }

    //printing the transpose matrix

    printf("The transpose of your matrix is\n");

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            printf("%d\t",B\_285[i\_285][j\_285]);

            if(j\_285==(n\_285-1))

            printf("\n");

        }

    }

    count\_285=0;

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            if(B\_285[i\_285][j\_285]==A\_285[i\_285][j\_285])

                count\_285=count\_285+1;

        }

    }

    if(count\_285==m\_285\*n\_285)

    printf("\nThe matrix is symmetric\n");

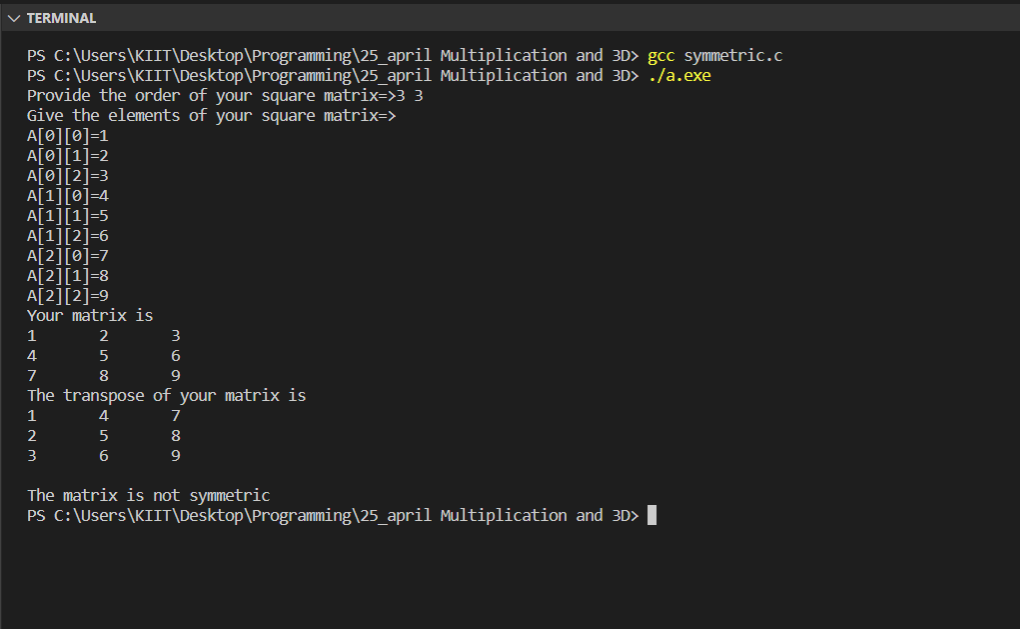
    else

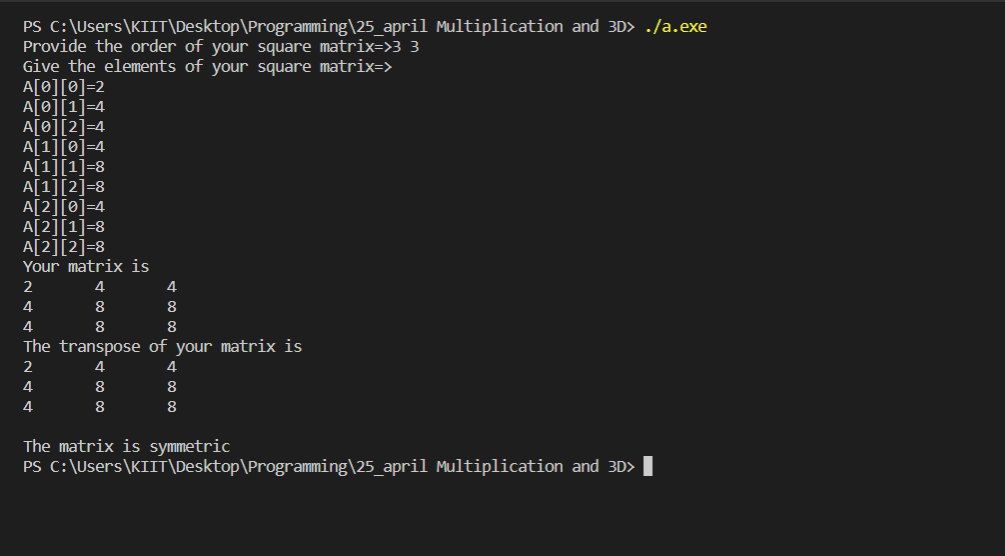
    printf("\nThe matrix is not symmetric");

    return 0;

}

Output:





#2. WAP to check whether a given matrix is orthogonal or not.

Code:

#include <stdio.h>

int main()

{

    int i\_285,j\_285,m\_285,k\_285,n\_285,A\_285[10][10],B\_285[10][10],C\_285[10][10],D\_285[10][10],I\_285[10][10],count\_285;

    printf("Provide the order of your matrix=>");

    scanf("%d%d",&m\_285,&n\_285);

    printf("Provide the elements of your matrix\n");

    //storing fresh input

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            printf("A[%d][%d]=",i\_285,j\_285);

            scanf("%d",&A\_285[i\_285][j\_285]);

        }

    }

    //identity matrix

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            if(i\_285==j\_285)

            I\_285[i\_285][j\_285]=1;

            else

            I\_285[i\_285][j\_285]=0;

        }

    }

    //storing the transpose in B

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            B\_285[i\_285][j\_285]=A\_285[j\_285][i\_285];

        }

    }

    //Multiplying A with B and storing in C

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            C\_285[i\_285][j\_285]=0;

            for(k\_285=0;k\_285<n\_285;k\_285++)

            {

                C\_285[i\_285][j\_285]+=A\_285[i\_285][k\_285]\*B\_285[k\_285][j\_285];

            }

        }

    }

        //Multiplying B with A and storing in D

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            D\_285[i\_285][j\_285]=0;

            for(k\_285=0;k\_285<n\_285;k\_285++)

            {

                D\_285[i\_285][j\_285]+=B\_285[i\_285][k\_285]\*A\_285[k\_285][j\_285];

            }

        }

    }

    count\_285=0;

    for(i\_285=0;i\_285<m\_285;i\_285++)

    {

        for(j\_285=0;j\_285<n\_285;j\_285++)

        {

            if(C\_285[i\_285][j\_285]==D\_285[i\_285][j\_285] && D\_285[i\_285][j\_285]==I\_285[i\_285][j\_285])

            {

                count\_285++;

            }

        }

    }

    if(count\_285==(m\_285\*n\_285))

    printf("Your matrix is orthogonal!\n");

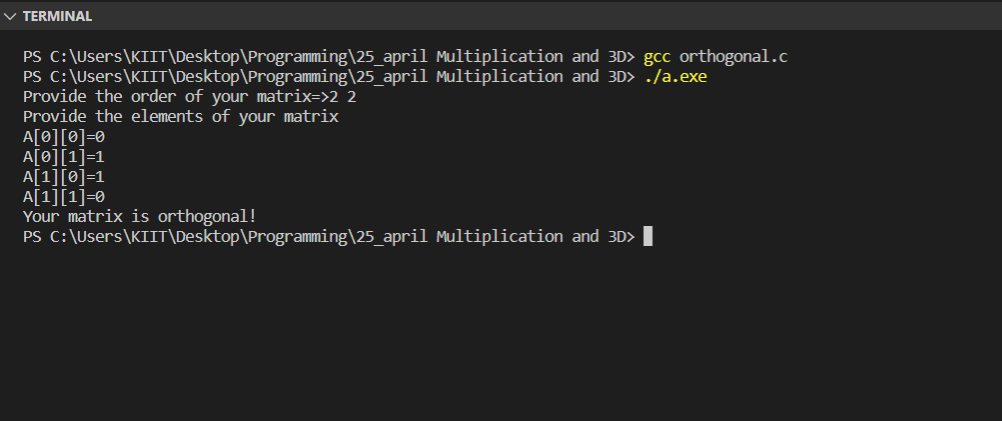
    else

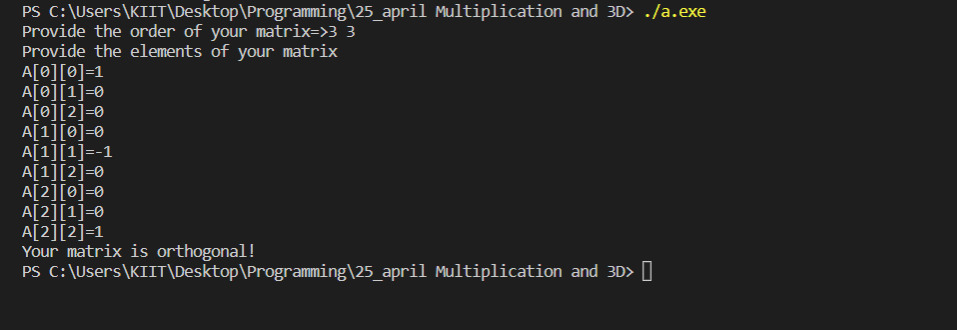
    printf("No it is not orthogonal!\n");

    return 0;

}

Output:





#3. Use functions for string copy, concatenate.

Code:

#include <stdio.h>

#include <string.h>

void main()

{

    char A\_285[20],B\_285[20];

    printf("Provide a short sentence\n");

    gets(A\_285);

    printf("Provide another short sentence\n");

    gets(B\_285);

    printf("Adding both of your sentences would give us===\n");

    strcat(A\_285,B\_285);

    puts(A\_285);

    printf("Over wrting your first sentence with the second would give us===\n");

    strcpy(A\_285,B\_285);

    puts(A\_285);

}

Output:

