

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)
```

KIDUS ABEBE MEKONEN

Roll No- 2106285

Section A11

IT department

```
printf("\nThe matrix is symmetric\n");
else
printf("\nThe matrix is not symmetric");
return 0;
}
```

Output:

```
▼ TERMINAL

PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> gcc symmetric.c
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> ./a.exe
Provide the order of your square matrix=>3 3
Give the elements of your square matrix=>
A[0][0]=1
A[0][1]=2
A[0][2]=3
A[1][0]=4
A[1][1]=5
A[1][2]=6
A[2][0]=7
A[2][1]=8
A[2][2]=9
Your matrix is
1      2      3
4      5      6
7      8      9
The transpose of your matrix is
1      4      7
2      5      8
3      6      9

The matrix is not symmetric
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> █
```

```
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> ./a.exe
Provide the order of your square matrix=>3 3
Give the elements of your square matrix=>
A[0][0]=2
A[0][1]=4
A[0][2]=4
A[1][0]=4
A[1][1]=8
A[1][2]=8
A[2][0]=4
A[2][1]=8
A[2][2]=8
Your matrix is
2      4      4
4      8      8
4      8      8
The transpose of your matrix is
2      4      4
4      8      8
4      8      8

The matrix is symmetric
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> █
```

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

Code:

```
#include <stdio.h>
int main()
{
```

KIDUS ABEBE MEKONEN

Roll No- 2106285

Section A11

IT department

```
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++)
```

KIDUS ABEBE MEKONEN

Roll No- 2106285

Section A11

IT department

```
{
    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:

```
▼ TERMINAL

PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> gcc orthogonal.c
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> ./a.exe
Provide the order of your matrix=>2 2
Provide the elements of your matrix
A[0][0]=0
A[0][1]=1
A[1][0]=1
A[1][1]=0
Your matrix is orthogonal!
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> █
```

```
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> ./a.exe
Provide the order of your matrix=>3 3
Provide the elements of your matrix
A[0][0]=1
A[0][1]=0
A[0][2]=0
A[1][0]=0
A[1][1]=-1
A[1][2]=0
A[2][0]=0
A[2][1]=0
A[2][2]=1
Your matrix is orthogonal!
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> █
```

#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);
```

KIDUS ABEBE MEKONEN

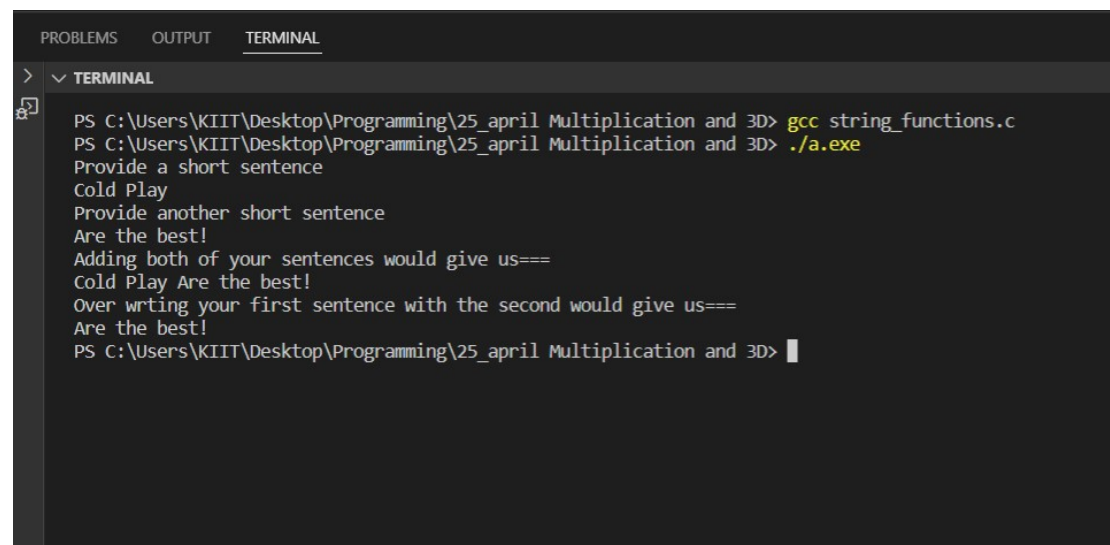
Roll No- 2106285

Section A11

IT department

```
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:



```
PROBLEMS  OUTPUT  TERMINAL
>  ✓ TERMINAL
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> gcc string_functions.c
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> ./a.exe
Provide a short sentence
Cold Play
Provide another short sentence
Are the best!
Adding both of your sentences would give us===
Cold Play Are the best!
Over wrting your first sentence with the second would give us===
Are the best!
PS C:\Users\KIIT\Desktop\Programming\25_april Multiplication and 3D> |
```