

DATA STRUCTURE

ASSIGNMENT NUMBER – 02

Aim: Accept conventional matrix and convert it into Sparse matrix. Implement Simple Transpose and Fast Transpose Algorithm on Sparse Matrix.

PROGRAM CODE:

```
#include<stdio.h>
void main()
{
    int i,j,k,m,n,p,q,sum,count=0;
    printf("\t\t\t\t\t WELCOME TO THE PROGRAM OF SPARSE MATRIX\n");
    printf("Enter the Number of Rows and Columns of the Matrix\n");
    scanf("%d %d",&m,&n);
    int a[m][n],b[m][n],c[m][n];
    printf("\nEnter the elements of the First Matrix:\n");
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < n; j++)
        {
            scanf("%d",&a[i][j]);
            if (a[i][j]==0)
            {
                count=count+1;
            }
        }
    }
}
```

```
printf("\nThe First Matrix is:\n\n");
for ( i = 0; i <m; i++)
{
    for ( j = 0; j < n; j++)
    {
        printf("      %d   ",a[i][j]);
    }
    printf("\n");
}
if (count>((m*n)/2))
{
    printf("\nNumber of Zeros is:%d\n",count);
    printf("Number of elements other than Zeros are:%d\n",(m*n)-count);
    printf("As number of Zeros are more than remaining elements\n");
    printf("\n\t\t\t\t\t\t\tTHE GIVEN MATRIX IS SPARSE MATRIX\n");
    printf("\nThe 3 Tuple Representation of SPARSE Matrix is:\n");
    printf("\nRows Columns Values");
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < n; j++)
        {
            if(a[i][j]!=0)
            {
                printf("\n      %d          %d          %d",i,j,a[i][j]);
            }
        }
        printf("\n");
    }
}
else
{
    printf("\nNumber of Zeros is:%d\n",count);
    printf("Number of elements other than Zeros are:%d\n",(m*n)-count);
    printf("As number of Zeros are less than remaining elements\n");
    printf("\n\t\t\t\t\t\t\tTHE GIVEN MATRIX IS DENSE MATRIX\n");
    printf("\nThe 3 Tuple Representation of DENSE Matrix is:\n");
    printf("\nRows Columns Values");
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < n; j++)
        {
            if(a[i][j]!=0)
            {
                printf("\n      %d          %d          %d\n",i,j,a[i][j]);
            }
        }
        printf("\n");
    }
}
```

```
printf("Enter the Number of Rows and Columns of the Matrix\n");
scanf("%d %d",&p,&q);
//int a[p][q],b[p][q],c[p][q];
printf("\nEnter the elements of the Second Matrix:\n");
for ( i = 0; i < p; i++)
{
    for ( j = 0; j < q; j++)
    {
        scanf("%d",&b[i][j]);
        if (b[i][j]==0)
        {
            count=count+1;
        }
    }
}
printf("\nThe Second Matrix is:\n\n");
for ( i = 0; i < p; i++)
{
    for ( j = 0; j < q; j++)
    {
        printf("    %d  ",b[i][j]);
    }
    printf("\n");
}
for ( i = 0; i < p; i++)
{
    for ( j = 0; j < q; j++)
    {
        c[i][j]=a[i][j]+b[i][j];
    }
}
if (m==p && n==q)
{
    printf("\nTHE MATRIX IS SQUARE MATRIX \n");
    printf("\nYOU CAN DO THE ADDITION OF TWO MATRICES\n");
    printf("\nTHE SUM OF TWO MATRICES IS:\n");
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < n; j++)
        {
            printf("    %d  ",c[i][j]);
        }
        printf("\n");
    }
}
else
{
    printf("\nYOU CANNOT DO THE ADDITION OF TWO MATRICES\n");
}
```

```
printf("\nAS THE MATRIX IS NOT SQUARE MATRIX\n");
}

if (n!=p)
{
    printf("\nMULTIPLICATION OF TWO MATRICES IS NOT POSSIBLE\n");
}
else
{
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < q; j++)
        {
            sum=0;
            for ( k = 0; k < m; k++)
            {
                sum=sum+(a[i][k]*b[k][j]);
            }
            c[i][j]=sum;
        }
    }
    printf("\nMULTIPLICATION OF TWO MATRICES IS:\n");
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; j < q; j++)
        {
            printf("    %d\t",c[i][j]);
        }
        printf("\n");
    }
}
```

Name: Siddhesh Dilip Khairnar
Roll no: 272028 (B)
PRN No: 22110398

OUTPUT:

The screenshot shows the Visual Studio Code interface with the file `Matrix_Representation.c` open. The code defines a function `main()` that prompts the user to enter the number of rows and columns of a matrix. In the terminal, the user has entered `2 2`. The program then asks for the elements of the first matrix, and the user has entered `1 0 0 0`. The output indicates that the matrix is sparse because the number of zeros (3) is greater than the number of non-zero elements (1). The 3-tuple representation of the sparse matrix is shown as `0 0 1`.

```
129 {
130     sum=0;
131     for ( k = 0; k < m; k++)
132     {
133         sum=sum+(a[i][k]*b[k][j]);
134     }
135     c[i][j]=sum;
136 }
137
138 }
139
140 printf("\nMULTIPLICATION OF TWO MATRICES IS:\n");
141 for ( i = 0; i < m; i++)
142 {
143     for ( j = 0; j < q; j++)
144     {
145         printf(" %d\t", c[i][j]);
146     }
147 }
```

WELCOME TO THE PROGRAM OF SPARSE MATRIX

Enter the Number of Rows and Columns of the Matrix
2 2

Enter the elements of the First Matrix:
1 0 0 0

The First Matrix is:

```
1 0
0 0
```

Number of Zeros is:3
Number of elements other than zeros are:1
As number of Zeros are more than remaining elements

THE GIVEN MATRIX IS SPARSE MATRIX

The 3 Tuple Representation of SPARSE Matrix is:

```
Rows Columns Values
0 0 1
```

The screenshot shows the same Visual Studio Code interface, but now the user has entered `2 2` for the number of rows and columns of the second matrix. The program then asks for the elements of the second matrix, and the user has entered `1 2 3 4`. The output indicates that the matrix is square (2x2) and that the user can perform addition of two matrices. The sum of the two matrices is shown as `2 2 3 4`. The multiplication of the two matrices is also shown, resulting in `1 2 0 0`.

```
129 {
130     sum=0;
131     for ( k = 0; k < m; k++)
132     {
133         sum=sum+(a[i][k]*b[k][j]);
134     }
135     c[i][j]=sum;
136 }
137
138 }
139
140 printf("\nMULTIPLICATION OF TWO MATRICES IS:\n");
141 for ( i = 0; i < m; i++)
142 {
143     for ( j = 0; j < q; j++)
144     {
145         printf(" %d\t", c[i][j]);
146     }
147 }
```

Enter the Number of Rows and Columns of the Matrix
2 2

Enter the elements of the Second Matrix:
1 2 3 4

The Second Matrix is:

```
1 2
3 4
```

THE MATRIX IS SQUARE MATRIX

YOU CAN DO THE ADDITION OF TWO MATRICES

THE SUM OF TWO MATRICES IS:

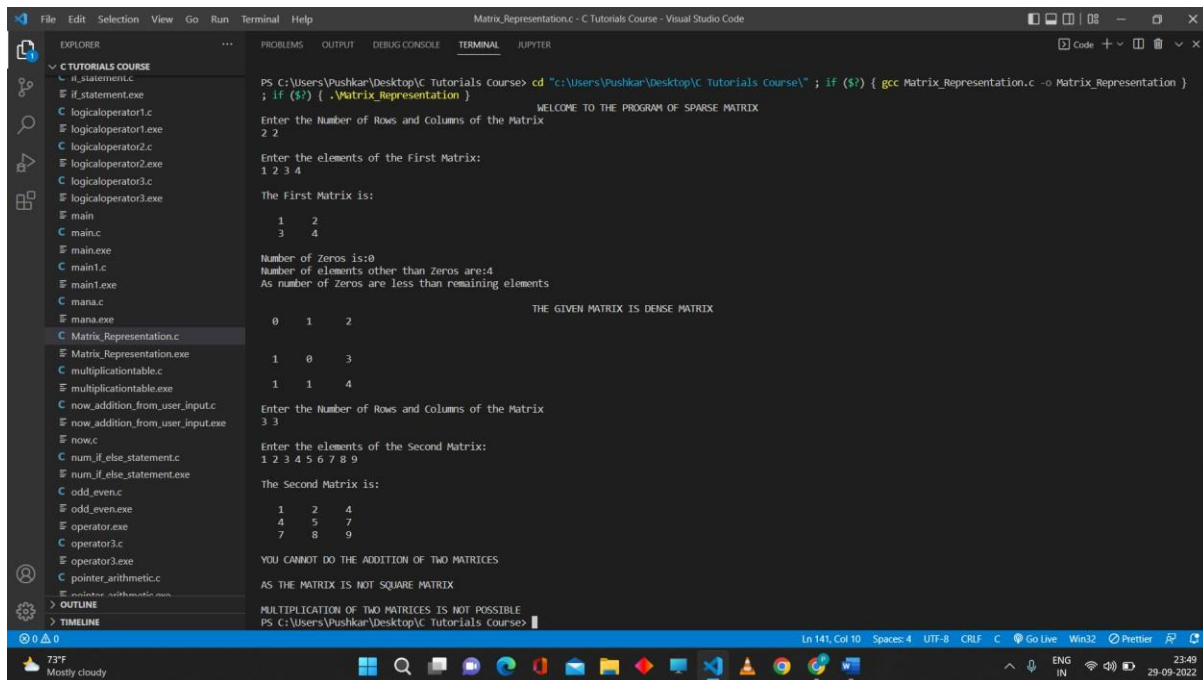
```
2 2
3 4
```

MULTIPLICATION OF TWO MATRICES IS:

```
1 2
0 0
```

PS C:\Users\Pushkar\Desktop\C Tutorials Course>

Name: Siddhesh Dilip Khairnar
Roll no: 272028 (B)
PRN No: 22110398



```
PS C:\Users\Pushkar\Desktop\C Tutorials Course> cd "c:\Users\Pushkar\Desktop\C Tutorials Course\" ; if ($?) { gcc Matrix_Representation.c -o Matrix_Representation }
; if ($?) { .\Matrix_Representation }

WELCOME TO THE PROGRAM OF SPARSE MATRIX
Enter the Number of Rows and Columns of the Matrix
2 2
Enter the elements of the First Matrix:
1 2 3 4
The First Matrix is:
1 2
3 4
Number of Zeros is:0
Number of elements other than Zeros are:4
As number of Zeros are less than remaining elements
THE GIVEN MATRIX IS DENSE MATRIX
0 1 2
1 0 3
1 1 4
Enter the Number of Rows and Columns of the Matrix
3 3
Enter the elements of the Second Matrix:
1 2 3 4 5 6 7 8 9
The Second Matrix is:
1 2 4
4 5 7
7 8 9
YOU CANNOT DO THE ADDITION OF TWO MATRICES
AS THE MATRIX IS NOT SQUARE MATRIX
MULTIPLICATION OF TWO MATRICES IS NOT POSSIBLE
PS C:\Users\Pushkar\Desktop\C Tutorials Course>
```

Conclusion: Thus, we have successfully executed the program for Sparse Matrix and also able to do the 3 Tuple Representation as well as Addition and Multiplication of the Two Matrices.