Program

```c
#include<stdio.h>
int create(int r,int c,int t[][3]){
    int m[r][c],i,j,k=0;
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            scanf("%d",&m[i][j]);
        }
    }
    t[0][0]=r;
    t[0][1]=c;
    for(i=0;i<r;i++){
        for(j=0;j<c;j++){
            if(m[i][j]!=0)
            {
                k++;
                t[k][0]=i;
                t[k][1]=j;
                t[k][2]=m[i][j];
            }
        }
    }
    t[0][2]=k;
    return k;
}
void display(int k,int t[][3]){
    int i;
    for(i=0;i<=k;i++)
    {
        printf("%d\t%d\t%d\n",t[i][0],t[i][1],t[i][2]);
    }
}
void addmatrix(int r,int c,int k1,int k2,int t1[][3],int t2[][3],int t3[][3]){
    int i=1,j=1,k=0;
    t3[0][0]=r;
    t3[0][1]=c;
    while(i<=k1&&j<=k2)
    {
        if(t1[i][0]==t2[j][0])
        {
            if(t1[i][1]==t2[j][1])
            {
                k++;
                t3[k][0]=t1[i][0];
                t3[k][1]=t1[i][1];
                t3[k][2]=t1[i][2]+t2[j][2];
                i++;
                j++;
            }
            else if(t1[i][1]<t2[j][1])
            {
                k++;
                t3[k][0]=t1[i][0];
                t3[k][1]=t1[i][1];
                t3[k][2]=t1[i][2];
                i++;
            }
            else
            {
                k++;
                t3[k][0]=t2[j][0];
```

Experiment No: 3
Date:

SPARSE MATRIX ADDITION

Aim:

To write a program to perform sparse matrix addition.

Algorithm:

1. Start
2. Define function create(int r,int c,int t[][3])

```
Declare matrix m[r][c] and integer variables i,j and k=0.
For i=0 to r do
     For j=0 to c do
        Read value of m[i][j]
Initialise t[0][0]=r
Initialise t[0][1]=c
For i=0 to r do
    For j=0 to c do
        If m[i][j]!=0
             k++
             t[k][0]=i;
             t[k][1]=j;
             t[k][2]=m[i][j]
Set t[0][2]=k;
Return k
```

3. Define function display(int k,int t[][3])

```
For i=0 to k do
      Print t[i][0],t[i][1],t[i][2]
```

4. Define function addmatrix(int r,int c,int k1,int k2,int t1[][3],int t2[][3],int t3[][3])

```
Initialise i=1,j=1,k=0
Set t3[0][0]=r and t3[0][1]=c
while(i<=k1&&j<=k2)

    if(t1[i][0]==t2[j][0])
        if(t1[i][1]==t2[j][1])
            k++
            t3[k][0]=t1[i][0]
            t3[k][1]=t1[i][1]
```

```
                t3[k][1]=t2[j][1];
                t3[k][2]=t2[j][2];
                j++;
            }
        }
        else if(t1[i][0]<t2[j][0])
        {
            k++;
            t3[k][0]=t1[i][0];
            t3[k][1]=t1[i][1];
            t3[k][2]=t1[i][2];
            i++;
        }
        else
        {
            k++;
            t3[k][0]=t2[j][0];
            t3[k][1]=t2[j][1];
            t3[k][2]=t2[j][2];
            j++;
        }
    }
    while(i<=k1)
    {
        k++;
        t3[k][0]=t1[i][0];
        t3[k][1]=t1[i][1];
        t3[k][2]=t1[i][2];
        i++;
    }
    while(j<=k2)
    {
        k++;
        t3[k][0]=t2[j][0];
        t3[k][1]=t2[j][1];
        t3[k][2]=t2[j][2];
        j++;
    }
    t3[0][2]=k;
    printf("\nRESULTANT MATRIX");
    for(i=0;i<=k;i++)
    {
        printf("\n%d\t%d\t%d",t3[i][0],t3[i][1],t3[i][2]);
    }
}
int main()
{
    int r1,c1,r2,c2,k1,k2,t1[50][3],t2[50][3],t3[50][3];
    printf("Enter the number of rows in matrix 1 : ");
    scanf("%d",&r1);
    printf("Enter the number of columns in matrix 1 : ");
    scanf("%d",&c1);
    printf("Enter the number of rows in matrix 2 : ");
    scanf("%d",&r2);
    printf("Enter the number of terms in matrix 2 : ");
    scanf("%d",&c2);
    if(r1!=r2||c1!=c2)
    {
        printf("\nAddition not possible");
        return 0;
    }
    printf("\nEnter matrix 1\n");
    k1=create(r1,c1,t1);
```

```
            t3[k][2]=t1[i][2]+t2[j][2]
            i++
            j++
        else if(t1[i][1]<t2[j][1])
            k++;
            t3[k][0]=t1[i][0]
            t3[k][1]=t1[i][1]
            t3[k][2]=t1[i][2]
            i++
        else
            k++;
            t3[k][0]=t2[j][0]
            t3[k][1]=t2[j][1]
            t3[k][2]=t2[j][2]
            j++

    else if(t1[i][0]<t2[j][0])
        k++;
        t3[k][0]=t1[i][0]
        t3[k][1]=t1[i][1]
        t3[k][2]=t1[i][2
        i++

    else
       k++
       t3[k][0]=t2[j][0]
       t3[k][1]=t2[j][1]
       t3[k][2]=t2[j][2]
       j++
while(i<=k1)
    k++
    t3[k][0]=t1[i][0]
    t3[k][1]=t1[i][1]
    t3[k][2]=t1[i][2]
    i++
while(j<=k2)
    k++
    t3[k][0]=t2[j][0]
    t3[k][1]=t2[j][1]
    t3[k][2]=t2[j][2]
    j++
Set t3[0][2]=k
Print resultant matrix
End addmatrix function
```

5. In main function ()

```
Read row value r1,r2 and column value c1,c2 of matrix 1 and 2
if(r1!=r2||c1!=c2)
    Print ‘‘Addition not possible’’
else
    Read elements of matrix 1
    Call function create k1=create(r1,c1,t1)
    Print sparse representation of matrix 1
    Read elements of matrix 2
    Call function create k2=create(r2,c2,t2)
    Print sparse representation of matrix 1
    Call function addmatrix(r1,c2,k1,t1,t2,t3)
```

6. Stop

```
    printf("\nSparse matrix representation of matrix 1\n");
    display(k1,t1);
    printf("\nEnter matrix 2\n");
    k2=create(r2,c2,t2);
    printf("\nSparse matrix representation of matrix 2\n");
    display(k2,t2);
    addmatrix(r1,c2,k1,k2,t1,t2,t3);
    return 0;
}
```

Output

```
Enter the number of rows in matrix 1 : 3
Enter the number of columns in matrix 1 : 4
Enter the number of rows in matrix 2 : 3
Enter the number of terms in matrix 2 : 4
Enter matrix 1
0
0
0
0
5
1
3
0
0
0
0
Sparse matrix representation of matrix 1
3 4 3
1 0 5
1 1 1
1 2 3
Enter matrix 2
6
0
0
5
0
4
6
0
0
0
0
7
Sparse matrix representation of matrix 2
3 4 5
0 0 6
0 3 5
1 1 4
1 2 6
2 3 7
RESULTANT MATRIX
3 4 6
0 0 6
0 3 5
1 0 5
1 1 5
1 2 9
2 3 3
```

Result:

Program has been executed successfully and obtained the output