

```

#include<stdio.h>
void main()
{
    int a[10][10],at[10][10],b[10][10],bt[10][10],ct[10][10];
    int i,j,k1,k2,r1,c1,r2,c2,m,n,k,r,c;
    printf("Enter the no. of rows and columns of Sparse Matrix A\n");
    scanf("%d %d",&r1,&c1);
    printf("Enter the elements\n");
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the no. of rows and columns of Sparse Matrix B\n");
    scanf("%d %d",&r2,&c2);
    printf("Enter the elements\n");
    for(i=0;i<r2;i++)
    {
        for(j=0;j<c2;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    printf("Sparse Matrix A\n");
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    printf("Sparse Matrix B\n");
    for(i=0;i<r2;i++)
    {
        for(j=0;j<c2;j++)
        {
            printf("%d ",b[i][j]);
        }
        printf("\n");
    }
    k1=1;
    for(i=0;i<r1;i++)
    {
        for(j=0;j<c1;j++)
        {
            if(a[i][j]!=0)
            {
                at[k1][0]=i;
                at[k1][1]=j;
                at[k1][2]=a[i][j];
                k1++;
            }
        }
    }
}

```



```

at[0][0]=r1;
at[0][1]=c1;
at[0][2]=k1-1;
k2=1;
for(i=0;i<r2;i++)
{
    for(j=0;j<c2;j++)
    {
        if(b[i][j]!=0)
        {
            bt[k2][0]=i;
            bt[k2][1]=j;
            bt[k2][2]=b[i][j];
            k2++;
        }
    }
}
bt[0][0]=r2;
bt[0][1]=c2;
bt[0][2]=k2-1;
printf("Tuple matrix A\n");
for(i=0;i<k1;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d ",at[i][j]);
    }
    printf("\n");
}
printf("Tuple matrix B\n");
for(i=0;i<k2;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d ",bt[i][j]);
    }
    printf("\n");
}
if(r1!=r2 || c1!=c2)
{
    printf("Addition is not possible\n");
}
else
{
    m=1;
    n=1;
    k=1;
    while(m<=at[0][2] && n<=bt[0][2])
    {
        if((at[m][0]==bt[n][0]) && (at[m][1]==bt[n][1]))
        {
            ct[k][0]=at[m][0];
            ct[k][1]=at[m][1];
            ct[k][2]=at[m][2]+bt[n][2];
            m++;
            n++;
            k++;
        }
    }
}

```



```

    }
    else if(at[m][0]==bt[n][0] && at[m][1]<bt[n][1])
    {
        ct[k][0]=at[m][0];
        ct[k][1]=at[m][1];
        ct[k][2]=at[m][2];
        m++;
        k++;
    }
    else if(at[m][0]==bt[n][0] && at[m][1]>bt[n][1])
    {
        ct[k][0]=bt[n][0];
        ct[k][1]=bt[n][1];
        ct[k][2]=bt[n][2];
        n++;
        k++;
    }
    else if(at[m][1]<bt[n][1])
    {
        ct[k][0]=at[m][0];
        ct[k][1]=at[m][1];
        ct[k][2]=at[m][2];
        m++;
        k++;
    }
    else if(at[m][1]>bt[n][1])
    {
        ct[k][0] = bt[n][0];
        ct[k][1] = bt[n][1];
        ct[k][2] = bt[n][2];
        n++;
        k++;
    }
}
while(m<=at[0][2])
{
    ct[k][0]=at[m][0];
    ct[k][1]=at[m][1];
    ct[k][2]=at[m][2];
    m++;
    k++;
}
while(n<=bt[0][2])
{
    ct[k][0]=bt[n][0];
    ct[k][1]=bt[n][1];
    ct[k][2]=bt[n][2];
    n++;
    k++;
}
ct[0][0]=at[0][0];
ct[0][1]=at[0][1];
ct[0][2]=k-1;
printf("Tuple of Sum Matrix\n");
for(i=0;i<k;i++)
{
    for(j=0;j<3;j++)

```



```

        {
            printf("%d ",ct[i][j]);
        }
        printf("\n");
    }
    r=ct[0][0];
    c=ct[0][1];
    k=1;
    printf("Sum Matrix\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            if(ct[k][0]==i && ct[k][1]==j)
            {
                printf("%d ",ct[k][2]);
                k++;
            }
            else
                printf("0 ");
        }
        printf("\n");
    }
}

```

