

```

#include <stdio.h>
#define MAX 100
typedef struct{
    int row;
    int col;
    int value;
}SPARSE;
SPARSE sparse1[MAX],sparse2[MAX],sum[MAX];
int m1,n1,m2,n2;
void toSparse(int matrix1[][n1],int matrix2[][n2]){
    int k1=0,k2=0,i,j;
    sparse1[0].row=sparse2[0].row=m1;
    sparse1[0].col=sparse2[0].col=n1;
    for (i = 0; i < m1; i++)
    for (j = 0; j < n1; j++){
        if(matrix1[i][j]){
            k1++;
            sparse1[k1].row = i ;
            sparse1[k1].col = j ;
            sparse1[k1].value = matrix1[i][j];
        }
        if(matrix2[i][j]){
            k2++;
            sparse2[k2].row = i ;
            sparse2[k2].col = j ;
            sparse2[k2].value = matrix2[i][j];
        }
    }
    sparse1[0].value=k1;
    sparse2[0].value=k2;
}
void calculateSum(){
    int k=1,i=1,j=1;
    sum[0].row=sparse1[0].row;
    sum[0].col=sparse1[0].col;
    while (i<=sparse1[0].value&&
           j<=sparse2[0].value){
        if (sparse1[i].row==sparse2[j].row&&
            sparse1[i].col==sparse2[j].col){
            sum[k].row = sparse1[i].row;
            sum[k].col = sparse1[i].col;
            sum[k].value = sparse1[i].value+
                           sparse2[j].value;
            i++;j++;
        }
        else if (sparse1[i].row==sparse2[j].row){
            sum[k].row = sparse1[i].row;
            if (sparse1[i].col < sparse2[j].col){
                sum[k].col = sparse1[i].col;
                sum[k].value = sparse1[i].value;
                i++;
            }
            else{
                sum[k].col = sparse2[j].col;
                sum[k].value = sparse2[j].value;
                j++;
            }
        }
        else
            if(sparse1[i].row < sparse2[j].row){
                sum[k].row = sparse1[i].row;
                sum[k].col = sparse1[i].col;
                sum[k].value = sparse1[i].value;
                i++;
            }
            else{
                sum[k].row = sparse2[j].row;
                sum[k].col = sparse2[j].col;
                sum[k].value = sparse2[j].value;
                j++;
            }
    }
    k++;
}
while(i<=sparse1[0].value){
    sum[k].row = sparse1[i].row;
    sum[k].col = sparse1[i].col;
    sum[k].value = sparse1[i].value;
    k++;i++; }
while(j<=sparse2[0].value){
    sum[k].row = sparse2[j].row;
    sum[k].col = sparse2[j].col;
    sum[k].value = sparse2[j].value;
    k++;j++; }
sum[0].value=k-1;
}
void display(SPARSE sparse[MAX]){
    printf("\nROW COLUMN VALUE");
    for (int i=0; i <= sparse[0].value;i++)
        printf("\n%d\t%d\t%d", sparse[i].row,
               sparse[i].col, sparse[i].value);
}
void readMatrix(int m,int n,int arr[][n]){
    printf("ENTER MATRIX\n");
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            scanf("%d",&arr[i][j]);
}
int main(){
    int matrix1[100][100],matrix2[100][100];
    printf("ENTER MATRIX_1
           DIMENSIONS :");
    scanf("%d%d",&m1,&n1);
    readMatrix(m1,n1,matrix1);
    printf("\nENTER MATRIX_2
           DIMENSIONS :");
    scanf("%d%d",&m2,&n2);
    if (m1!=m2||n1!=n2)
        printf("\nINVALID DIMENSION
               FOR ADDITION\n");
    else{
        readMatrix(m2,n2,matrix2);
        toSparse(matrix1,matrix2);
        printf("MATRIX_1 IN TUPLE
               FORM");
        display(sparse1);
        printf("\nMATRIX_2 IN TUPLE
               FORM");
        display(sparse2);
        calculateSum();
        printf("\nSUM IN TUPLE FORM");
        display(sum);
    }
}

```