

Sparse Matrix using C

```
#include <stdio.h>
#include<stdlib.h>

struct Element
{
    int i;
    int j;
    int x;
};

struct Sparse
{
    int m;
    int n;
    int num;
    struct Element *ele;
};

void create(struct Sparse *s)
{
    int i;

    printf("Enter Dimensions");
    scanf("%d%d",&s->m,&s->n);
    printf("Number of non-zero");
    scanf("%d",&s->num);

    s->ele=(struct Element *)malloc(s->num*sizeof(struct
Element));
    printf("Enter non-zero Elements");
    for(i=0;i<s->num;i++)
        scanf("%d%d%d",&s->ele[i].i,&s->ele[i].j,&s-
>ele[i].x);
}

void display(struct Sparse s)
{
    int i,j,k=0;

    for(i=0;i<s.m;i++)
    {
        for(j=0;j<s.n;j++)
```

```

        {
            if(i==s.ele[k].i && j==s.ele[k].j)
                printf("%d ",s.ele[k++].x);
            else
                printf("0 ");
        }
        printf("\n");
    }
}

struct Sparse * add(struct Sparse *s1,struct Sparse *s2)
{
    struct Sparse *sum;
    int i,j,k;
    i=j=k=0;

    if(s1->n != s2->n && s1->m != s2->m)
        return NULL;
    sum=(struct Sparse *)malloc(sizeof(struct Sparse));
    sum->ele=(struct Element *)malloc((s1->num+s2->num)*sizeof(struct Element));

    while(i<s1->num && j<s2->num)
    {
        if(s1->ele[i].i<s2->ele[j].i)
            sum->ele[k++]=s1->ele[i++];
        else if(s1->ele[i].i>s2->ele[j].i)
            sum->ele[k++]=s2->ele[j++];
        else
        {
            if(s1->ele[i].j<s2->ele[j].j)
                sum->ele[k++]=s1->ele[i++];
            else if(s1->ele[i].j>s2->ele[j].j)
                sum->ele[k++]=s2->ele[j++];
            else
            {
                sum->ele[k]=s1->ele[i];
                sum->ele[k++].x=s1->ele[i++].x+s2->ele[j+
+].x;
            }
        }
    }
    for(;i<s1->num;i++)sum->ele[k++]=s1->ele[i];
    for(;j<s2->num;j++)sum->ele[k++]=s2->ele[j];
    sum->m=s1->m;
    sum->n=s1->n;
    sum->num=k;
}

```

```
        return sum;
    }

int main()
{
    struct Sparse s1,s2,*s3;

    create(&s1);
    create(&s2);

    s3=add(&s1,&s2);

    printf("First Matrix\n");
    display(s1);
    printf("Second Matrix\n");
    display(s2);
    printf("Sum Matrix\n");
    display(*s3);

    return 0;
}
```