

```

#include<stdio.h>
#include<stdlib.h>
typedef struct SET
{
    float nr[10];
    float dr[10];
    int n;
}fuzzy;

fuzzy *getval(fuzzy *m,int ch,char *x)
{
    int i;
    float f,g;
    m=(fuzzy*)malloc(sizeof(fuzzy*));
    m->n=ch;
    printf("\n enter the set %s",x);
    for(i=0;i<m->n;i++)
    {
        printf("numerator element %d",i+1);
        scanf("%f",&f);
        m->nr[i]=f;
        printf("denominator element %d",i+1);
        scanf("%f",&g);
        m->dr[i]=g;
    }
    return(m);
}

void printval(fuzzy *m,char *x)
{
    int i;
    float f;
    printf("\n The set %s=(",x);
    for(i=0;i<m->n;i++)
    {
        printf("%6.1f / %6.1f",m->nr[i],m->dr[i]);
        if(i!=m->n-1)
            printf(",");
    }
    printf(")");
}

fuzzy *algebraic_product(fuzzy *a,fuzzy *b)
{
    fuzzy *temp;
    temp=(fuzzy*)malloc(sizeof(fuzzy*));
    int i;
    temp->n=a->n;
    for(i=0;i<a->n;i++)
    {
        temp->nr[i]=a->nr[i]*b->nr[i];
        temp->dr[i]=a->dr[i];
    }
    return temp;
}

fuzzy *algebraic_sum(fuzzy *a,fuzzy *b)
{
    fuzzy *temp;

```

```

    int i;
    temp->n=a->n;
    for(i=0;i<a->n;i++)
    {
        temp->nr[i]=(a->nr[i]+b->nr[i])-(a->nr[i]*b->nr[i]);
        temp->dr[i]=a->dr[i];
    }
    return temp;
}

void cartesian_product(fuzzy *a,fuzzy *b)
{
    float prod[10][10];
    int j;
    for(i=0;i<a->n;i++)
    {
        for(j=0;j<b->n;j++)
        {
            if(a->nr[i]<b->nr[j])
                prod[i][j]=a->nr[i];
            else
                prod[i][j]=b->nr[j];
            printf("%d\t",prod[i][j]);
        }
        printf("\n");
    }
}

void main()
{
    fuzzy *a=NULL;
    fuzzy *b=NULL;
    fuzzy *result1,*result2,*result3;
    int c,ch;
    printf("\n enter the no.of components");
    scanf("%d",&c);

    a=getval(a,c,"A");
    b=getval(b,c,"B");
    printval(a,"A");
    printval(b,"B");
    do
    {
        printf("\n1.Algebraic sum \n 2.Algebraic Product \n 3.Cartesian
Product");
        printf("\n enter your choice");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1: result=algebraic_sum(a,b);
                    printval(result,"D");
                    break;
            case 2: result=algebraic_product(a,b);
                    printval(result,"E");
                    break;
            case 3: result=cartesian_product(a,b);
                    break;
            default:break;
        }
    }
}

```

```
}while(ch!=3)
```