1. /*Write a program that takes two or more sets as input and produces set operations like union, intersection, difference and symmetric difference as its output.*/

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

int set_union(int setA[], int m, int setB[], int n, int
UNION[])
{

int i,j,k=0;

for(i=0;i<m;i++){
     UNION[k]=setA[i];
     k++;
}

for(i=0;i<n;i++){
     int flag=1;
     for(j=0;j<m;j++){
          if(setA[j]== setB[i]){
               flag = 0;
               Break;
          }
     }
     if(flag ==1){
          UNION[k] = setB[i];
          k++;
     }
     }
     return (k);
}


int Intersection(int setA[],int m,int setB[],int n,int
INTER[]){
     int i,j,k=0,l;
     for(i=0;i<m;i++){
          for(j=0;j<n;j++){
                    if(setA[i]==setB[j]){
                         INTER[k]=setA[i];
                    k++;
               }
          }
     }
     return k;
}
```

```
int Difference(int setA[],int m, int setB[],int n,int DIFF[]){
     int i,j,k=0;
     for(i=0;i<m;i++){
     int flag=0;
     for(j=0;j<n;j++){
          if(setA[i]==setB[j]){
               flag =1;
               break;
          }
     }
     if (flag == 0){
          DIFF[k++]=setA[i];
     }
     }
     return k;
}


int Symmetric(int setA[],int m,int setB[],int n,int SYMM[]){
     int k = set_union(setA,m,setB,n,SYMM);
}

int element(int set[],int size){
     int i;
     for(i=0;i<size;i++){
          scanf("%d",&set[i]);
     }
}

void display(int set[],int size){
     int i,j,k;
     printf("{");
     for(i=0;i<size;i++){
          if(i<size-1){
               printf("%d, ",set[i]);
          }
          else if (i==size-1){
               printf("%d",set[i]);
          }
     }
     printf("}\n\n");
}

void bubblesort(int set[], int size){

     int i, j,temp;
     for(i=0;i<size-1;i++){
          for(j=0;j<size-i-1;j++){
               if ( set[j]>set[j+1]){
```

```
                    temp = set[j+1];
                    set[j+1] = set[j];
                    set[j] = temp;
                }
            }
        }
}

int main(){
    int m,n,i,j,UNION[40],INTERSECTION[40],
    DIFFA[40],DIFFB[40], SYM[40], INTER[40],SYMM[40];

    int union_num,inter_num,diff_numA,diff_numB,symm_num;

    printf("Enter size of set A: ");
    scanf("%d",&m);
    printf("Enter size of set B: ");
    scanf("%d",&n);

    int setA[m], setB[n];


    printf("Enter the elements of setA: \n");

    element(setA, m);
    bubblesort(setA,m);

    printf("Enter the elements of setB: \n");
    element(setB, n);
    bubblesort(setB,n);

    //displaying the elemensts of set A and set B
    printf("The Elements of the setA: \n");
    display(setA,m);
    printf("The Elements of the setB: \n");
    display(setB,n);

    //union set operation
    printf("The union of setA and setB: \n");
    union_num = set_union(setA,m,setB,n,UNION);
    bubblesort(UNION,union_num);
    display(UNION,union_num);

    //intersection set operation
    printf("The intersection of setA and setB: \n");
    inter_num = Intersection(setA,m,setB,n,INTER);

    bubblesort(INTER,inter_num);
    display(INTER,inter_num);
```

```
        //difference set operation
        printf("The difference of setA and setB (SetA - SetB):
        \n");
        diff_numA = Difference(setA,m,setB,n,DIFFA);
        bubblesort(DIFFA,diff_numA);
        display(DIFFA, diff_numA);

        //difference set operation
        printf("The difference of setA and setB (SetB - SetA):
        \n");
        diff_numB = Difference(setB,m,setA,n,DIFFB);
        bubblesort(DIFFB,diff_numB);
        display(DIFFB,diff_numB);

             //symmetric difference set opration
        printf("The symmetric difference of setA and setB ((SetA
        - SetB) U (SetB - SetA)) : \n");
        symm_num = Symmetric(DIFFA,diff_numA,DIFFB,diff_numB,
        SYMM);
        bubblesort(SYMM,symm_num);
        display(SYMM, symm_num);

}
```

OUTPUT:

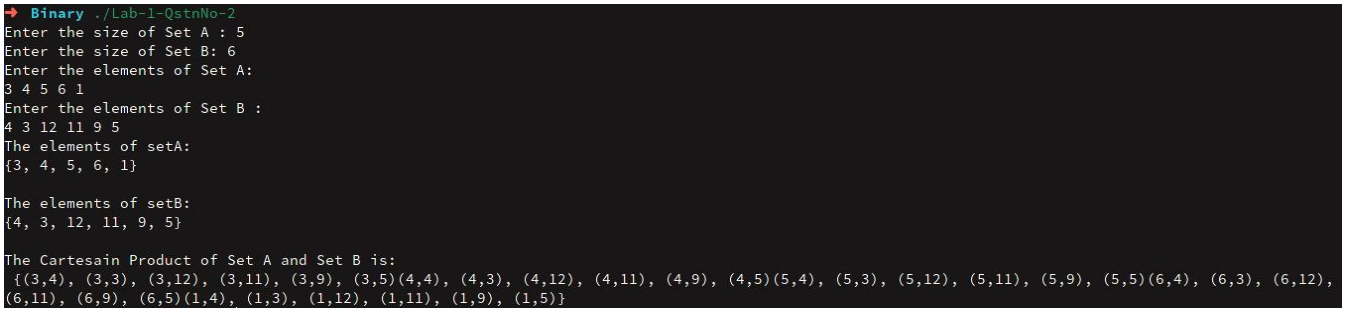
```
➜ Binary ./Lab-1-QstnNo-1
Enter size of set A: 5
Enter size of set B: 6
Enter the elements of setA:
4 5 6 7 10
Enter the elements of setB:
3 4 10 11 29 3
The Elements of the setA:
{4, 5, 6, 7, 10}

The Elements of the setB:
{3, 3, 4, 10, 11, 29}

The union of setA and setB:
{3, 3, 4, 5, 6, 7, 10, 11, 29}

The intersection of setA and setB:
{4, 10}

The difference of setA and setB (SetA - SetB):
{5, 6, 7}

The difference of setA and setB (SetB - SetA):
{3, 3, 11}

The symmetric difference of setA and setB ((SetA - SetB) U (SetB - SetA)) :
{3, 3, 5, 6, 7, 11}
```

2. /*Write a program that takes two or more sets as input and produces their Cartesian product as output. */

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


void CartProduct(int arr1[],int arr2[],int n1, int n2){

     int i,j;
     printf("{");
     for(i=0; i < n1; i++){
          for(j=0; j < n2; j++){
                    if (j<n2-1){
                         printf("(%d,%d), ",arr1[i],arr2[j]);
                    }
                    else if (j== n2-1){
                         printf("(%d,%d)",arr1[i],arr2[j]);
                    }
          }

     }
     printf("}\n");
}

void display(int set[],int size){
     int i,j,k;
     printf("{");
     for(i=0;i<size;i++){
          if(i<size-1){
               printf("%d, ",set[i]);
          }
          else if (i==size-1){
               printf("%d",set[i]);
          }
     }
     printf("}\n\n");
}
int main(){

int n1,n2,i,j;
printf("Enter the size of Set A : ");
scanf("%d",&n1);

printf("Enter the size of Set B: ");
scanf("%d",&n2);
int setA[n1],setB[n2];

printf("Enter the elements of Set A:\n");
```

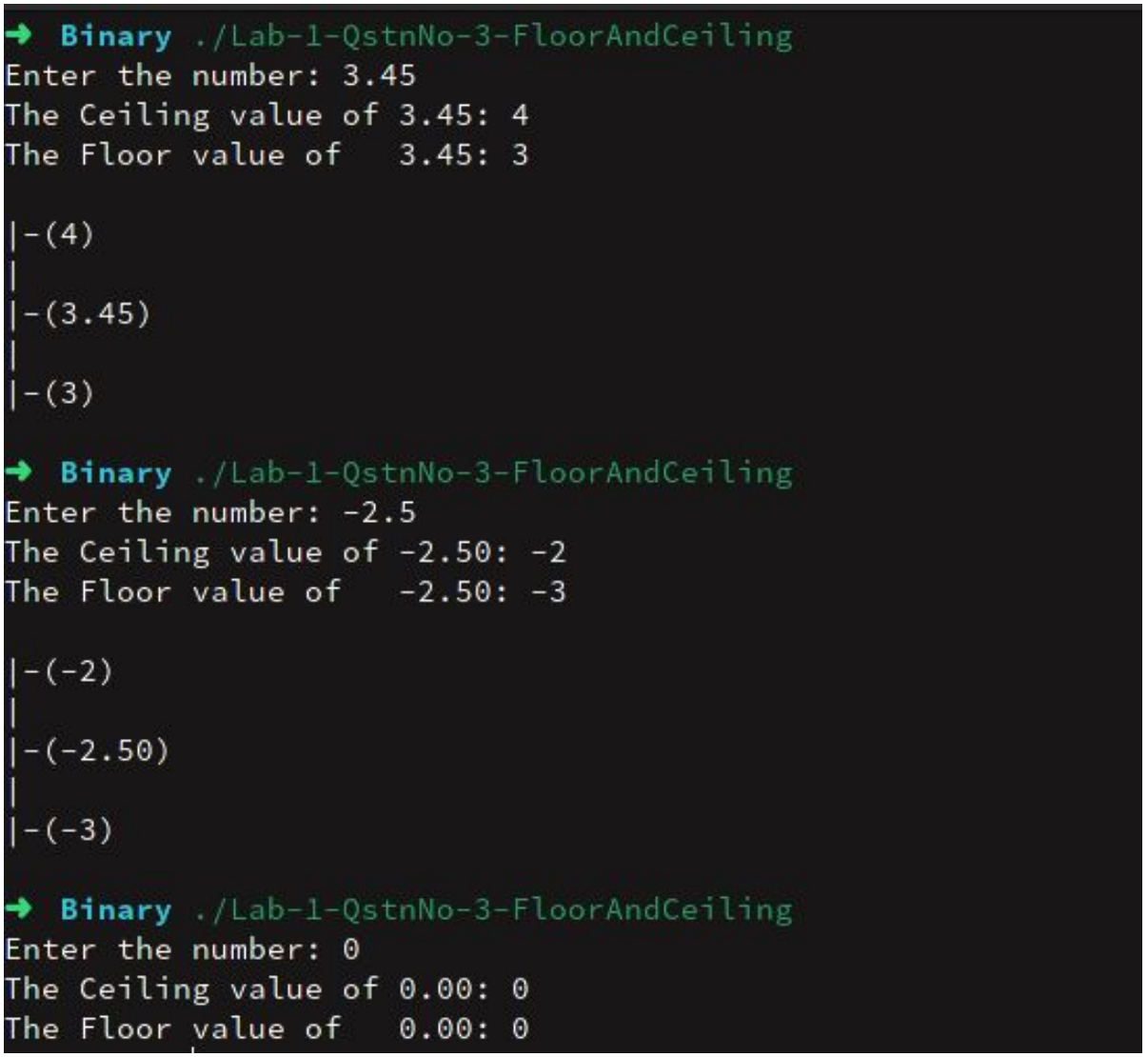
```
for(i=0; i<n1; i++){
     scanf("%d",&setA[i]);
}

printf("Enter the elements of Set B :\n");
for(j=0;j<n2;j++){
     scanf("%d",&setB[j]);

}
printf("The elements of setA: \n");
display(setA,n1);
printf("The elements of setB: \n");
display(setB,n2);

printf("The Cartesain Product of Set A and Set B is: \n ");

CartProduct(setA,setB,n1,n2);
return 0;
}
```

OUTPUT:



3. /*Write a program that takes a real number and produces is ceiling and floor integers as output. */

```
#include<stdio.h>
int floorValue(float num){
     if (num == (int)(num)){
          return num;
     }
     else if (num>=0){
          return (int)(num/1);
     }
     else{
          return (int)(num-1);
     }
}

int CeilValue(float num){
     if(num == (int)(num)){
          return num;
     }
     else if(num>=0){
          return (int)((num/1)+1);
     }
     else{
          return (int)(num);
     }
}

int main(){

     float x;
     int ceil,floor;
     printf("Enter the number: ");
     scanf("%f",&x);

     ceil = CeilValue(x);

     floor = floorValue(x);

     printf("The Ceiling value of %.2f: %d\n",x,ceil);
     printf("The Floor value of   %.2f: %d\n", x ,floor);

     if ( (int)(x) != x)
     {
          printf("\n|-(%d)\n|\n|-(%.2f)\n|\n|-(%d)\n\n",ceil,x
     ,floor);
     }
}
```


➜ Binary ./Lab-1-QstnNo-3-FloorAndCeiling
Enter the number: 3.45
The Ceiling value of 3.45: 4
The Floor value of 3.45: 3
|-(4)
|
|-(3.45)
|
|-(3)
➜ Binary ./Lab-1-QstnNo-3-FloorAndCeiling
Enter the number: -2.5
The Ceiling value of -2.50: -2
The Floor value of -2.50: -3
|-(-2)
|
|-(-2.50)
|
|-(-3)
➜ Binary ./Lab-1-QstnNo-3-FloorAndCeiling
Enter the number: 0
The Ceiling value of 0.00: 0
The Floor value of 0.00: 0

4. /*Write a program that takes name and age of a 5 persons as an input and gives the degree of membership of the person as its output according to following membership functions.
   a. Degree of membership = 1 if age<=20
      Degree of membership = (30-age)/10 if age>20 and age<=30
      Degree of membership = 0 if age>30
   b. Degree of membership = 1 if age<=15
      Degree of membership = (35-age)/20 if age>15 and age<=35
      Degree of membership = 0 if age>35
   Perform set operations according to rules of fuzzy sets, on these two sets.
   */

```
#include<stdio.h>
#include<stdlib.h>
float degree_of_membershipA(int age){
     if (age <=20)       return 1;
     else if(age<=30)    return  (float)(30-age)/10;
     else                return 0;
}

float degree_of_membershipB(int age){
     if (age <=15)       return 1;
     else if(age<=35)    return  (float)(35-age)/20;
     else                return 0;
}

/****************************************Fuzzy
Union*****************************************/
void fuzzy_union(char Name[40][40],float MembershipA[40],
float MembershipB[40]){
     float union_Set[20];
     int i,j;
     for(i=0;i<5;i++)
     {
          if(MembershipA[i]>MembershipB[i])
          {
               union_Set[i]=MembershipA[i];
          }
          else if(MembershipA[i]< MembershipB[i])
          {
               union_Set[i]= MembershipB[i];
          }
          else
          {
```

```
            union_Set[i]=MembershipA[i];
          }
     }

        printf("The result of the union fuzzy operation is :
\n {");
     for(i=0;i<5;i++){
          if(i<4){
                printf("%.2f/%s,",union_Set[i],Name[i]);
          }
          else if(i == 4){
                printf("%.2f/%s",union_Set[i],Name[i]);

          }
     }
        printf("}\n\n");
}


/**************Fuzzy intersection**************/

void fuzzy_intersection(char Name[40][40],float
MembershipA[40], float MembershipB[40]){
     float intersection_set[20];
     int i,j;
     for(i=0;i<5;i++)
     {
          if(MembershipA[i]>MembershipB[i])
          {
                intersection_set[i]=MembershipB[i];
          }
          else if(MembershipA[i]< MembershipB[i])
          {
                intersection_set[i]= MembershipA[i];
          }
          else
          {
                intersection_set[i]=MembershipA[i];
          }
     }

        printf("The result of the intersection fuzzy operation
is : \n {");
     for(i=0;i<5;i++){
          if(i<4){
                printf("%.2f/%s,
",intersection_set[i],Name[i]);
          }
          else if(i==4){
                printf("%.2f/%s",intersection_set[i],Name[i]);
          }
```

```
     }
        printf("}\n\n");
}

/***************Fuzzy Complement********************/
void fuzzy_complement(char Name[40][40],float MembershipA[40],
float MembershipB[40]){
     float complement_SetA[20],complement_SetB[20];
     int i,j;
     for(i=0;i<5;i++)
     {
            complement_SetA[i]=1-MembershipA[i];
            complement_SetB[i]=1-MembershipB[i];
     }

printf("The result of the Complement fuzzy operation of first
set is : \n {");
     for(i=0;i<5;i++){
          if(i<4){
               printf("%.2f/%s, ",complement_SetA[i],Name[i]);
          }
          else if(i==4){
                printf("%.2f/%s",complement_SetA[i],Name[i]);
          }
     }
printf("}\n\n");

printf("The result of the Complement fuzzy operation of second
set is : \n {");
     for(i=0;i<5;i++){
          if(i<4){
                printf("%.2f/%s, ",complement_SetB[i],
Name[i]);
          }
          else if(i==4){
                printf("%.2f/%s",complement_SetB[i],Name[i]);
          }
     }
     printf("}\n\n");
}


int main(){
     int age[40],i=0;
     char name[40][40];
     float membershipA[20],membershipB[20];
     for(i=0;i<5;i++){
     printf("Enter the name: ");     scanf("%s",name[i]);
     printf("Enter age: ");          scanf("%d",&age[i]);

     }
```

```
for(i=0;i<5;i++){
   membershipA[i]= degree_of_membershipA(age[i]);
   membershipB[i]= degree_of_membershipB(age[i]);
}

printf("First Set is: \n {");
for(i=0;i<5;i++){
     if(i<4){
            printf("%.2f/%s, ",membershipA[i],name[i]);
     }
     else if(i==4){
            printf("%.2f/%s",membershipA[i],name[i]);
     }
}
printf("}\n\n");

printf("Second Set is: \n {");
for(i=0;i<5;i++){
     if(i<4){
            printf("%.2f/%s, ",membershipB[i],name[i]);
     }
     else if(i==4){
            printf("%.2f/%s",membershipB[i],name[i]);
     }
}
printf("}\n\n");

fuzzy_union(name, membershipA, membershipB);
fuzzy_intersection(name,membershipA,membershipB);
fuzzy_complement(name,membershipA,membershipB);

}
```

OUTPUT:

```
➜  Binary ./Lab-1-QstnNo-4-fuzzyOperations
Enter the name: a
Enter age: 25
Enter the name: b
Enter age: 20
Enter the name: c
Enter age: 14
Enter the name: d
Enter age: 30
Enter the name: e
Enter age: 34
First Set is:
 {0.50/a, 1.00/b, 1.00/c, 0.00/d, 0.00/e}

Second Set is:
 {0.50/a, 0.75/b, 1.00/c, 0.25/d, 0.05/e}

The result of the union fuzzy operation is :
 {0.50/a,1.00/b,1.00/c,0.25/d,0.05/e}

The result of the intersection fuzzy operation is :
 {0.50/a, 0.75/b, 1.00/c, 0.00/d, 0.00/e}

The result of the Complement fuzzy operation of first set is :
 {0.50/a, 0.00/b, 0.00/c, 1.00/d, 1.00/e}

The result of the Complement fuzzy operation of second set is :
 {0.50/a, 0.25/b, 0.00/c, 0.75/d, 0.95/e}
```