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.

**Code**:

#include<stdio.h>

int a[10],b[10],c[10],d[10],i,j,k=0,n,m,flag=0;

void unio(){

for(i=0;i<n;i++){

c[k]=a[i];

k++; }

for(i=0;i<m;i++){

flag=0;

for(j=0;j<n;j++){

if(b[i]==a[j]){

flag=1;

break;

}

}

if(flag==0){

c[k]=b[i];

k++;

}

}

printf("\n Union \n");

for(i=0;i<k;i++){

printf("%d ",c[i]);

}

}

void intersection(){

printf("\nIntersections\n");

for(i=0;i<n;i++){

for(j=0;j<m;j++){

if(a[i]==b[j])

printf("%d ",a[i]);

}

}

}

void difference(){

printf("\nA-B\n");

for(i=0;i<n;i++){

flag=0;

for(j=0;j<m;j++){

if(a[i]==b[j]){

flag=1;

break;

}

}

if(flag==0)

printf("%d ",a[i]);

}

printf("\n\nB-A\n");

for(i=0;i<m;i++){

flag=0;

for(j=0;j<n;j++){

if(b[i]==a[j]){

flag=1;

break;

}

}

if(flag==0)

printf("%d ",b[i]);

}

}

void symmetric\_diff(){

k=0;

for(i=0;i<n;i++){

flag=0;

for(j=0;j<m;j++){

if(a[i]==b[j]){

flag=1;

break;

}

}

if(flag==0){

d[k]=a[i];

k++;

}

}

for(i=0;i<m;i++){

flag=0;

for(j=0;j<n;j++){

if(b[i]==a[j]){

flag=1;

break;

}

}

if(flag==0){

d[k]=b[i];

k++;

}

}

printf("\n(A-B)U(B-A)\n");

for(i=0;i<k;i++){

printf("%d ",d[i]);

}

}

int main(){

printf("Enter the size of array A\n");

scanf("%d",&n);

printf("Enter the element of First array A\n");

for(i=0;i<n;i++){

scanf("%d",&a[i]);

}

printf("Enter the size of array B\n");

scanf("%d",&m);

printf("Enter the elements of array B\n");

for(j=0;j<m;j++){

scanf("%d",&b[j]);

}

unio();

printf("\n");

intersection();

printf("\n");

printf("Difference of set");

difference();

printf("\n");

printf("Symmetric Difference");

symmetric\_diff();

printf("\n");

return 0;

}

**Output**

****

2. Write a program that takes two or more sets as input and produces their

Cartesian product as output.

**Code**:

#include<stdio.h>

int main(){

int a[50],b[50],c[50],i,s1,s2,j,k;

printf("Enter how many elements in set 1 :\n");

scanf("%d",&s1);

printf("Enter how many elements in set 2 :\n");

scanf("%d",&s2);

printf("Enter elements of set 1 :\n");

for(i=0;i<s1;i++) {

scanf("%d",&a[i]);

}

printf("Enter elements of set 2 :\n");

for(i=0;i<s2;i++){

scanf("%d",&b[i]);

}

printf("\tCartesian product are: ");

for(i=0;i<s1;i++){

for(j=0;j<s2;j++){

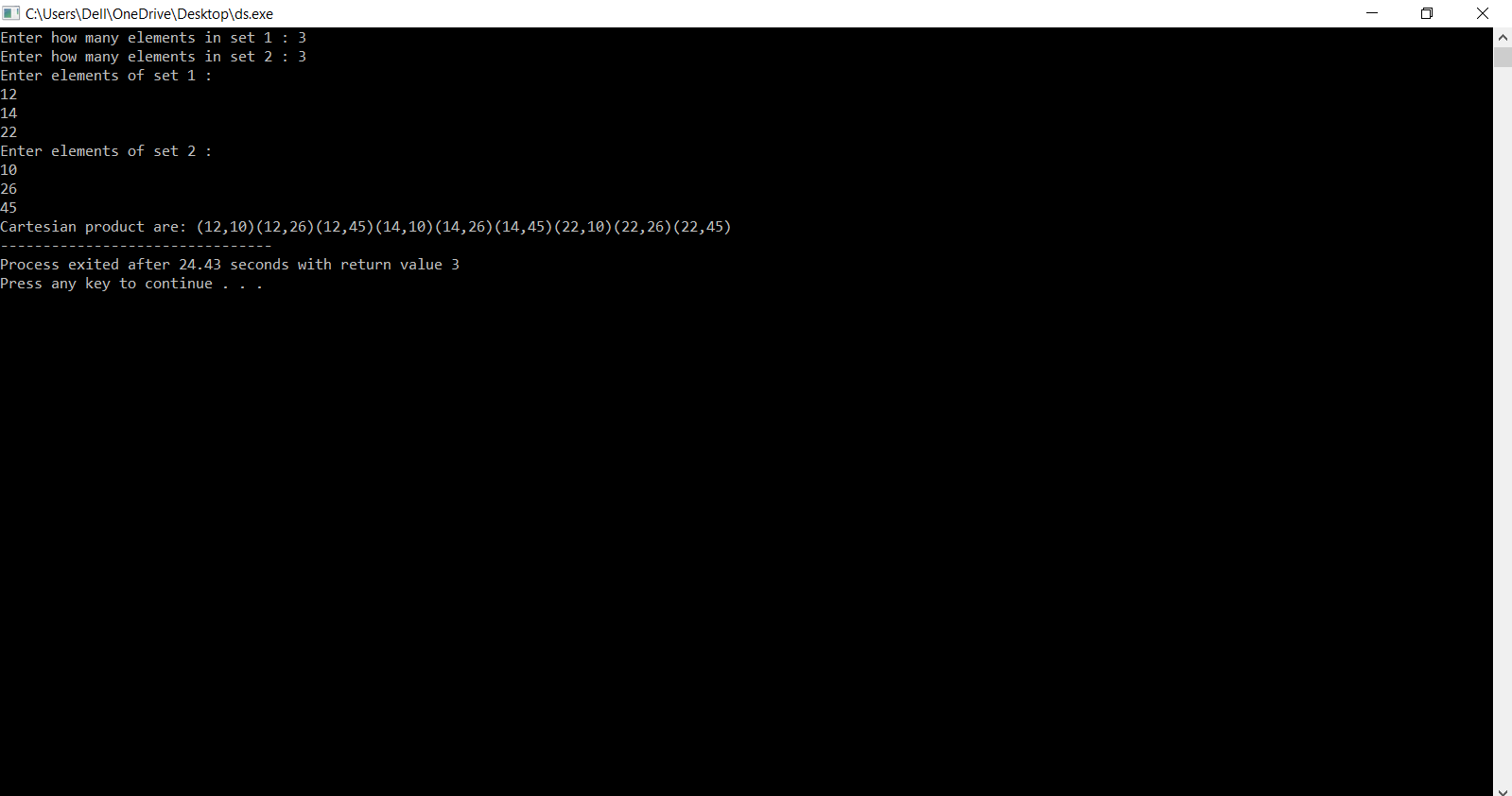
printf("(%d,%d)",a[i],b[j]);

}

}

}

**Output**



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

**Code**:

#include <stdio.h>

int floor(float x){

if (x == (int)(x)){

return x;

}

else if(x < 0){

return (int)(x)-1;

}

else{

return (int)(x);

}

}

int ceil(float x)

{

if(x == (int)(x))

{

return x;

}

else if(x >= 0)

{

return (int)(x)+1;

}

else{

return (int)(x);

}

}

int main(){

float num;

printf("Enter the value: ");

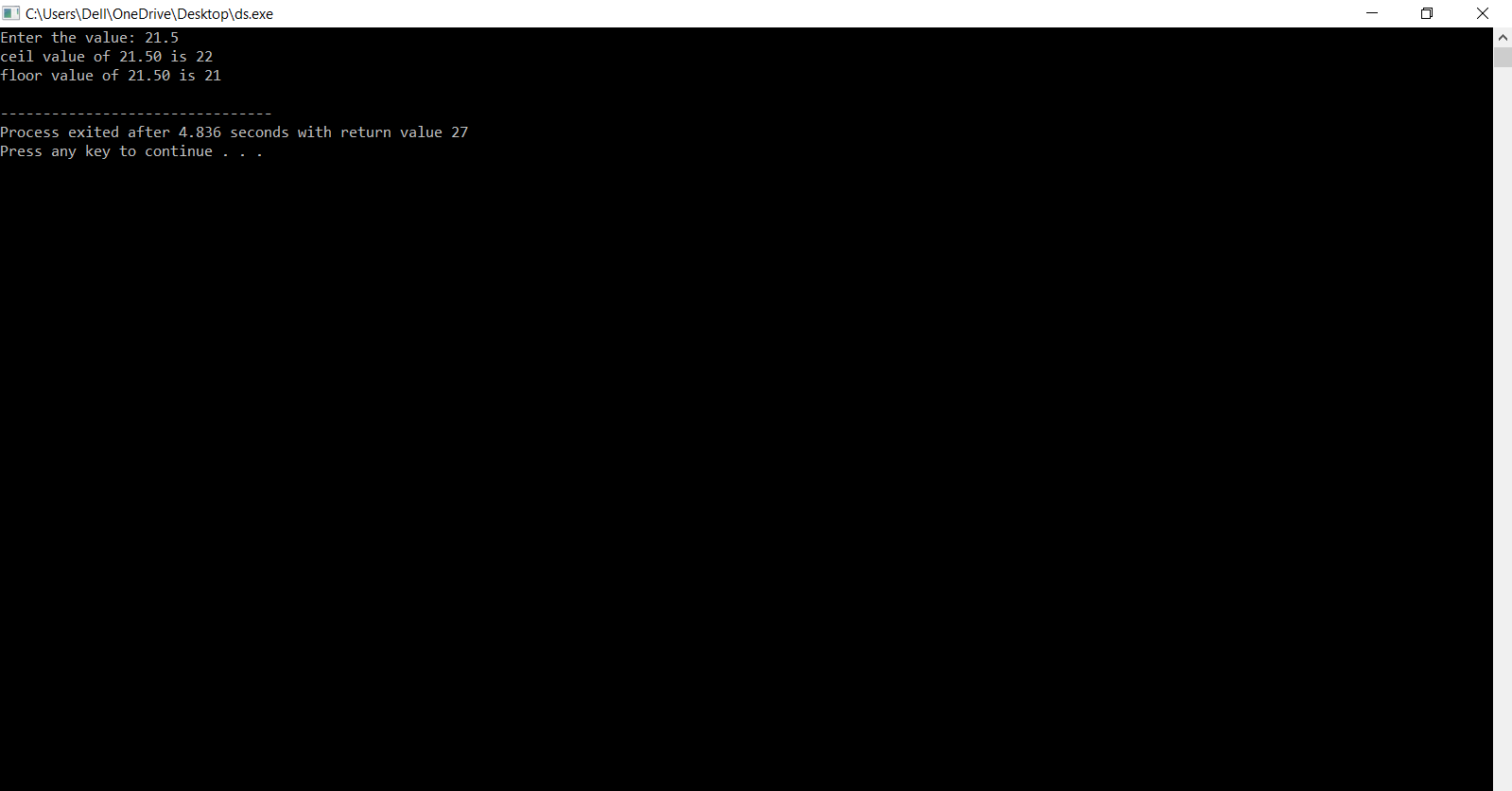
scanf("%f", &num);

printf("ceil value of %.2f is %d\n",num,ceil(num));

printf("floor value of %.2f is %d\n",num,floor(num));

}

**Output**



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

Degree of membership = 0

b. Degree of membership = 1

Degree of membership = (35-age)/20

Degree of membership = 0

two sets.

if age>20 and age<=30

if age>30

if age<=15

if age>15 and age<=35

if age>35

Perform set operations according to rules of fuzzy sets, on these

**Code**:

#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]);

}

system("clear");

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");

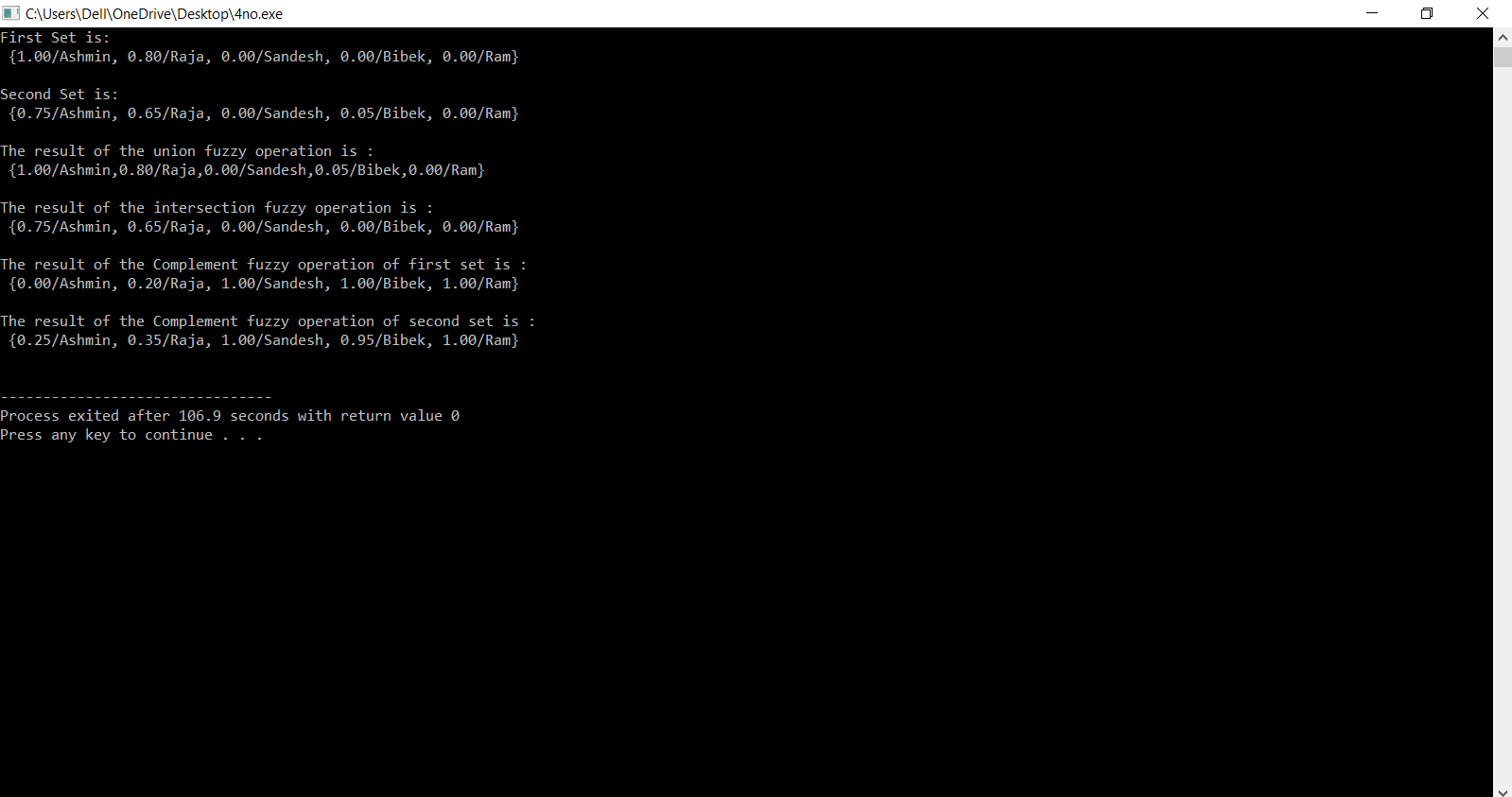
fuzzy\_union(name, membershipA, membershipB);

fuzzy\_intersection(name,membershipA,membershipB);

fuzzy\_complement(name,membershipA,membershipB);

}

**Output**

****