

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.

program:

```
#include <iostream>
#include <stdlib.h>
using namespace std;
void union(int all [10]) {
    cout << "{"
    for (int i=0; i<10; i++) {
        int flag = 0
        for (int j=i+1; j<10; j++) {
            if (all[i] == all[j])
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
            cout << " " << all[i];
        cout << " " << endl;
    }
}
void intersection(int all [10]) {
    cout << "{"
    for (int i=0; i<10; i++) {
        for (int j=i+1; j<10; j++)
        {
            if (all[i] == all[j] && i!=j)
            {
                cout << " " << all[i];
            }
        }
    }
    cout << " " << endl;
}
```

```

cout << "Enter for elements of  $\beta$  : " << endl;
for (int j = 0; j < 8; j++) {
    cout << "for element " << j+1 << " : ";
    cin >> b[j];
}
system("cls");
cout << "Set A is : " << endl << "{";
for (int i = 0; i < 5; i++) {
    cout << " " << a[i];
}
cout << "}" << endl;
cout << "Set  $\beta$  is : " << endl << "{";
for (int j = 0; j < 5; j++) {
    cout << " " << b[j];
}
cout << "}" << endl;
for (int i = 0; i < 5; i++) {
    all[i] = a[i];
}
for (int j = 0; j < 5; j++) {
    all[j+5] = b[j];
}
cout << "A  $\cup$   $\beta$  is : " << endl;
union(all);
cout << "A  $\cap$   $\beta$  is : " << endl;
intersection(all);
cout << "A -  $\beta$  is : " << endl;
difference(a, b);
cout << "A" << Char(30) << " $\beta$  is : " << endl;
sy difference(all);
return 0;
}

```

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

Program:

```
#include <iostream>
using namespace std;
int main() {
    int a[5], b[5], n, m;
    cout << "Enter no. of elements of set A : ";
    cin >> n;
    cout << "Enter for set A : " << endl;
    for (int i = 0; i < n; i++) {
        cout << "For element " << i + 1 << " : ";
        cin >> a[i];
    }
    cout << "Enter no. of elements for set B : ";
    cin >> m;
    cout << "Enter for set B : " << endl;
    for (int j = 0; j < m; j++) {
        cout << "For element " << j + 1 << " : ";
        cin >> b[j];
    }
    cout << "Cartesian product is : ";
    cout << " { ";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << "(" << a[i] << " " << b[j] << " ) " << " ";
        }
    }
    cout << " } ";
    return 0;
}
```

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

Program:

```
#include <iostream>
using namespace std;
float a;
int b;
cout << "Enter integer value: ";
cin >> a;
b = a;
if (a < 0) {
    cout << "Floor value of " << a << " is: " << b-1 << endl;
    cout << "Ceiling value of " << a << " is: " << b << endl;
}
if (a != b && a > 0) {
    cout << "Floor value of integer " << a << " is: " << b << endl;
    cout << "Ceiling value of " << a << " is: " << b+1 << endl;
}
if (a == b) {
    cout << "Floor value of " << a << " is: " << b << endl;
    cout << "Ceiling value of " << a << " is: " << b << endl;
}
return 0;
}
```

4. write a program that takes name and age of 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

a. Degree of membership = $(30 - \text{age})/10$ if age > 20 and age ≤ 30

Degree of membership = 0 if age > 30

b. Degree of membership = 1 if age ≤ 15

Degree of membership = $(35 - \text{age})/20$ if age > 15 and age ≤ 35

Degree of membership = 0 if age > 35

Perform set operations according to rules of fuzzy sets, on the two sets

program:

```
#include <iostream>
using namespace std;
int main() {
    string name[5];
    double age[5], membership[5], age1[5], membership1[5];
    cout << "Enter names: " << endl;
    for (int i = 0; i < 5; i++) {
        cout << "Name of " << i + 1 << " person: ";
        cin >> name[i];
    }
    cout << endl << "Enter age: " << endl;
    for (int i = 0; i < 5; i++) {
        if (age[i] = 20) {
            cout << "Degree of membership of " << name[i] << " = " <<
                membership[i] << endl;
            membership[i] = 1;
        }
    }
```



```

if (age[i] > 20 && age[i] <= 30) {
    cout << "Degree of membership of " << name[i] << " = " <<
        (30 - age[i]) / 10 << endl;
    membership[i] = (30 - age[i]) / 10;
}
if (age[i] > 30) {
    cout << "Degree of membership of " << name[i] << " = " << 0 << endl;
    membership[i] = 0;
}
}
cout << "Enter age for another set" << endl;
for (int i = 0; i < 5; i++) {
    cout << "Age of " << name[i] << ": ";
    cin >> age1[i];
}
for (int i = 0; i < 5; i++) {
    if (age1[i] <= 15) {
        cout << "Degree of membership of " << name[i] << ": " <<
            1 << endl;
        membership1[i] = 1;
    }
    if (age1[i] > 15 && age1[i] <= 35) {
        cout << "Degree of membership of " << name[i] << " = " <<
            (35 - age1[i]) / 20 << endl;
        membership1[i] = (35 - age1[i]) / 20;
    }
    if (age1[i] > 35) {
        cout << "Degree of membership of " << name[i] << " = " << 0;
        membership1[i] = 0;
    }
}
}

```

```
void difference(int a[5], int b[5]) {
```

```
    cout << " ";
```

```
    for (int i = 0; i < 5; i++) {
```

```
        int flag = 0;
```

```
        for (int j = 0; j < 5; j++) {
```

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

```
                flag = 1;
```

```
                break;
```

```
            }
```

```
        if (flag == 0) {
```

```
            cout << " " << a[i];
```

```
        }
```

```
    }
```

```
    cout << " " << endl;
```

```
}
```

```
void sydifference(int a1[10]) {
```

```
    cout << " ";
```

```
    for (int i = 0; i < 10; i++) {
```

```
        int flag = 0;
```

```
        for (int j = 0; j < 10; j++) {
```

```
            if (a1[i] == a1[j] && i != j) {
```

```
                flag = 1;
```

```
                break;
```

```
            }
```

```
        if (flag == 0) {
```

```
            cout << " " << a1[i];
```

```
        }
```

```
    }
```

```
    cout << " ";
```

```
}
```

```
int main() {
```

```
    int a[5], b[10], a1[10];
```

```
    cout << "Enter for set A: " << endl;
```

```
    for (int i = 0; i < 5; i++) {
```

```
        cout << "for element " << (i+1) << " : ";
```

```
        cin >> a[i];
```

```
    }
```

```

cout << "union of two fuzzy set is:" << endl << "1";
for (int i = 0; i < 5; i++) {
    if (membership[i] > membership1[i]) {
        cout << membership[i] << "1" << name[i] << ", ";
    }
    else {
        cout << membership1[i] << "1" << name[i] << ", ";
    }
}
cout << "}" << endl;
cout << "Intersection of two fuzzy set is" << endl << "1";
for (int i = 0; i < 5; i++) {
    if (membership[i] == membership1[i]) {
        cout << membership[i] << "1" << name[i] << ", ";
    }
}
cout << "}" << endl;
cout << "Compliment of 1st fuzzy set is" << endl << "1";
for (int i = 0; i < 5; i++) {
    cout << (1 - membership[i]) << "1" << name[i] << ", ";
}
cout << "}" << endl;
cout << "Complement of 2nd fuzzy set is" << endl << "1";
for (int i = 0; i < 5; i++) {
    cout << (1 - membership1[i]) << "1" << name[i] << ", ";
}
cout << "}" << endl;
return 0;
}

```