

**Lab 1**: WAP to find the intersection of two sets.

Intersection of Sets

The intersection of sets for two given sets is the set that contains all the elements that are common to both sets. The symbol for the intersection of sets is "∩''. For any two sets A and B, the intersection, A ∩ B (read as A intersection B) lists all the elements that are present in both sets, and are the common elements of A and B.

Intersection of sets is the set of elements which are common to both the given sets. In set theory, for any two sets A and B, the intersection is defined as the set of all the elements in set A that are also present in set B. We use the symbol '∩' that denotes 'intersection of'. For example, let us represent the students who like ice creams for dessert, Brandon, Sophie, Luke, and Jess. This is set A. The students who like brownies for dessert are Ron, Sophie, Mia, and Luke. This is set B. The students who like both ice creams and brownies are Sophie and Luke. This is represented as A ∩ B.

**Source Code**

*//wap to find the intersection of two sets*

#include<iostream>

using namespace std;

int main()

{

*//defining variables*

int a[100];

int b[100];

int c[100],k=0;

int n1,n2;

// input of set datas

cout<<"Enter the size of set a"<<endl;

cin>>n1;

cout<<"Enter the size of set b"<<endl;

cin>>n2;

*//input of set a*

cout<<"Enter the members of set a"<<endl;

for(int i=0;i<n1;i++){

cin>>a[i];

}

cout<<"A={";

for(int i=0;i<n1;i++){

if (i!= n1-1)

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

if (i == n1-1)

cout <<a[i];

}

cout<<"}"<<endl;

*//input of set b*

cout<<"Enter the members of set b"<<endl;

for(int i=0;i<n2;i++){

cin>>b[i];

}

cout<<"B={";

for(int i=0;i<n2;i++){

if (i!= n2-1)

cout<<b[i]<<" , ";

if (i == n2-1)

cout <<b[i];

}

cout<<"}"<<endl;

*//finding the intersection of set a and b*

for(int i=0;i<n1;i++){

for(int J=0;J<n2;J++)

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

c[k]=a[i];

cout<<c[k]<<endl;

k++;

}

}

*//printing of intersection of set a and b which is* c.

cout<<"Intersection is";

cout<<"C={";

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

if (i!= k-1)

cout<<c[i]<<" , ";

if (i == k-1)

cout <<c[i];

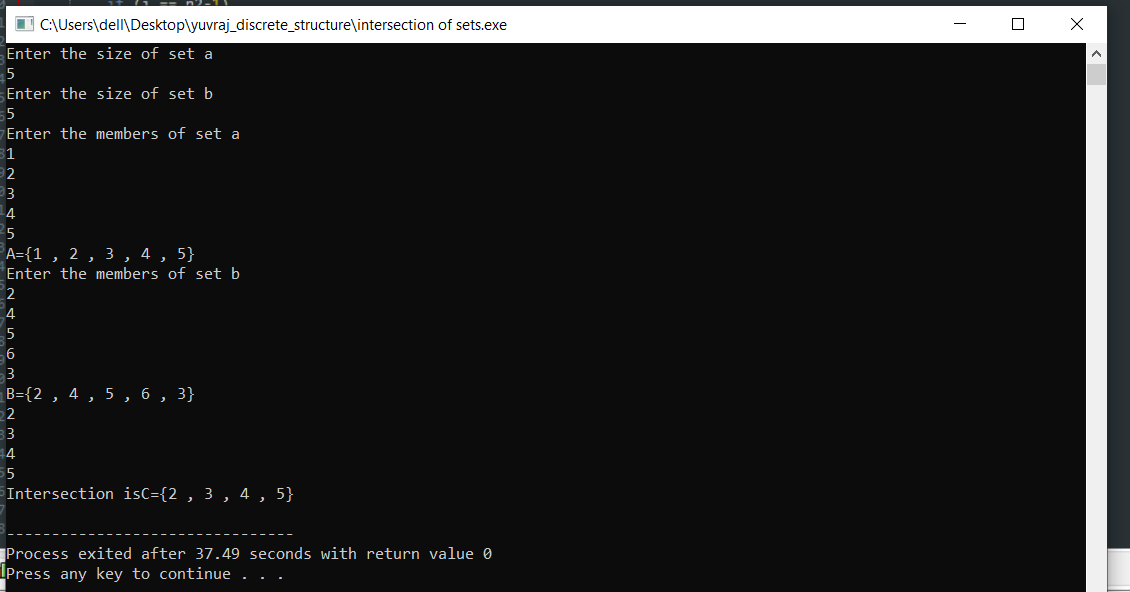
}

cout<<"}"<<endl;

return 0;

}

**Output**



**Lab 2** : WAP to find the union of two sets.

Union of Sets

The union of two sets X and Y is equal to the set of elements that are present in set X, in set Y, or in both the sets X and Y. This operation can be represented as;

X ∪ Y = {a: a ∈ X or a ∈ Y}

Let us consider an example, say;  set A = {1, 3, 5} and set B = {1, 2, 4} then;

A ∪ B = {1, 2, 3, 4, 5}

The union of set A and set B is equal to the set containing all the elements in A and B. This is represented as A U B and can be read as “A union B” or “A or B”.

A union B formula is generally used to calculate the unions of set A and set B. The formula for A union B indicates that each element present in A or B (leaving duplicates) is present in A U B. From the definition of the union of sets, the formula for A union B formula can be written as:

A U B = {x : x ∈ A or x ∈ B}

**Source Code**

*//wap to find the union of two sets*

#include<iostream>

using namespace std;

int main()

{

*//defining variables*

int a[100];

int b[100];

int c[100],k=0;

int p[100],q[100];

int n1,n2;

int f=0,d=0,e=0;

// input of set datas

cout<<"Enter the size of set a"<<endl;

cin>>n1;

cout<<"Enter the size of set b"<<endl;

cin>>n2;

//input of set a

cout<<"Enter the members of set a"<<endl;

for(int i=0;i<n1;i++){

cin>>a[i];

}

cout<<"A={";

for(int i=0;i<n1;i++){

if (i!= n1-1)

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

if (i == n1-1)

cout <<a[i];

}

cout<<"}"<<endl;

*//input of set b*

cout<<"Enter the members of set b"<<endl;

for(int i=0;i<n2;i++){

cin>>b[i];

}

cout<<"B={";

for(int i=0;i<n2;i++){

if (i!= n2-1)

cout<<b[i]<<" , ";

if (i == n2-1)

cout <<b[i];

}

cout<<"}"<<endl;

*//finding the intersection of set a and b*

for(int i=0;i<n1;i++){

for(int J=0;J<n2;J++)

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

c[k]=a[i];

k=k+1;

d=k;

}

}

e=n1+n2-d; *//no of members of set c*

for (int i = 0 ; i<n1 ; i++){

p[i]=a[i];

}

*//p is the set of whole set a and set b*

k=0;

for (int i = n1; i<n1+n2;i++)

{

p[i]=b[k];

k++;

}

*//q is the set of whole set a and b not containg intersection set*

k=0;

for ( int i = 0 ; i < n1+n2;i++ ){

f=0;

for (int j = 0;j<n1+n2-e;j++ ){

if (p[i]==c[j]){

f=1;}

}

if(f==0){

q[k]=p[i];

k++;

}

}

*//transferring data from q to c and making c the union set.*

k=0;

for (int i =n1+n2-e; i<e;i++){

c[i]=q[k];

k++;

}

*//printing of union of set a and b which is c.*

cout<<"Union is ";

cout<<"C={";

for(int i=0;i<e;i++){

if (i!= e-1)

cout<<c[i]<<" , ";

if (i == e-1)

cout <<c[i];

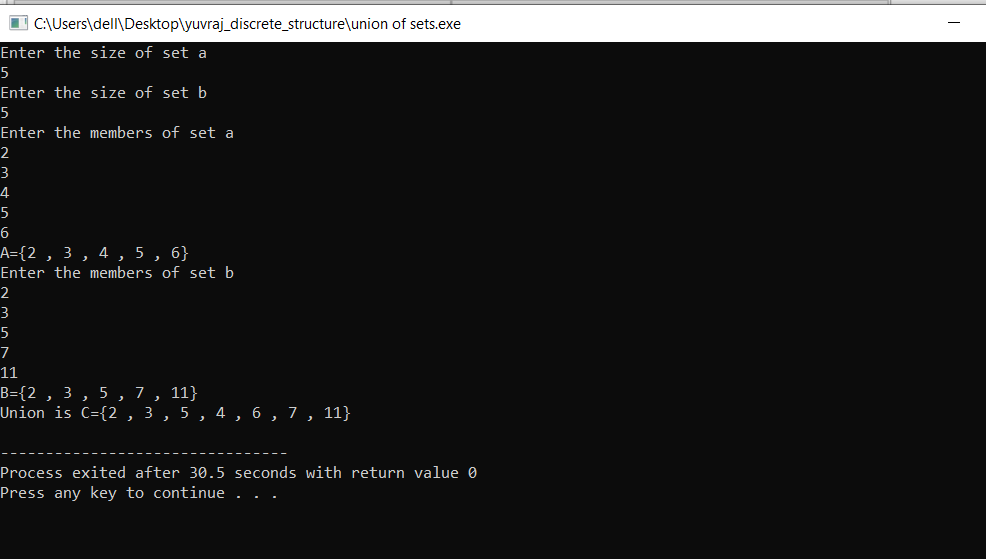
}

cout<<"}"<<endl;

return 0;

}

**Output**



**Lab 3**: WAP to find the difference of two sets

Difference of Sets

The difference of sets is one of the important and fundamental set theory operations. Union and intersection are the other set theory operations in addition to the difference of sets. The difference of two sets A and B is again a set that consists of the elements of A that are NOT in B.

The difference of two sets A and B is defined as the lists of all the elements that are in set A but that are not present in set B. The set notation used to represent the difference between the two sets A and B is A − B or A ∖ B. A - B in set-builder notation is defined as follows:

A - B = {x / x ∈ A and x ∉ B}

* A - B = the set that is obtained by removing the elements of A ∩ B from A
* B - A = the set that is obtained by removing the elements of A ∩ B from B

Consider the same example of two sets A = {1, 2, 3, 4, 5} and B = {3, 4, 5, 6, 7, 9}.We found that the difference A - B = {1, 2} and the difference B - A = {6, 7, 9}. Thus, we can see from this example that**A - B ≠ B - A.**

**Source Code**

*//wap to find the difference of two sets*

#include<iostream>

using namespace std;

int main()

{

*//defining variables*

int a[100];

int b[100];

int c[100],k=0;

int p[100],q[100];

int n1,n2;

int f=0,d=0,e=0;

// input of set datas

cout<<"Enter the size of set a"<<endl;

cin>>n1;

cout<<"Enter the size of set b"<<endl;

cin>>n2;

//input of set a

cout<<"Enter the members of set a"<<endl;

for(int i=0;i<n1;i++){

cin>>a[i];

}

cout<<"A={";

for(int i=0;i<n1;i++){

if (i!= n1-1)

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

if (i == n1-1)

cout <<a[i];

}

cout<<"}"<<endl;

*//input of set b*

cout<<"Enter the members of set b"<<endl;

for(int i=0;i<n2;i++){

cin>>b[i];

}

cout<<"B={";

for(int i=0;i<n2;i++){

if (i!= n2-1)

cout<<b[i]<<" , ";

if (i == n2-1)

cout <<b[i];

}

cout<<"}"<<endl;

*//finding the intersection of set a and b*

for(int i=0;i<n1;i++){

for(int J=0;J<n2;J++)

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

c[k]=a[i];

k=k+1;

d=k;

}

}

e=n1+n2-d; *//no of members of universal set*

d = n1+n2-e ;

*//q is the set of the difference of set a and b*

k=0;

for ( int i = 0 ; i < n1+n2;i++ ){

f=0;

for (int j = 0;j<n1+n2-e;j++ ){

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

f=1;}

}

if(f==0){

q[k]=a[i];

k++;

}

}

*//printing*

cout<<"Difference of a and b is ";

cout<<" A-B = {";

for(int i=0;i<e-n2;i++){

if (i!= e-n2-1)

cout<<q[i]<<" , ";

if (i == e-1-n2)

cout <<q[i];

}

cout<<"}"<<endl;

*// p is the set of difference of set b and set a*

k=0;

for ( int i = 0 ; i < n1+n2;i++ ){

f=0;

for (int j = 0;j<n1+n2-e;j++ ){

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

f=1;}

}

if(f==0){

p[k]=b[i];

k++;

}

}

*//printing*

cout<<"Difference of b and a is ";

cout<<" B-A = {";

for(int i=0;i<e-n1;i++){

if (i!= e-n1-1)

cout<<p[i]<<" , ";

if (i == e-1-n1)

cout <<p[i];

}

cout<<"}"<<endl;

return 0;

}

**Output**

