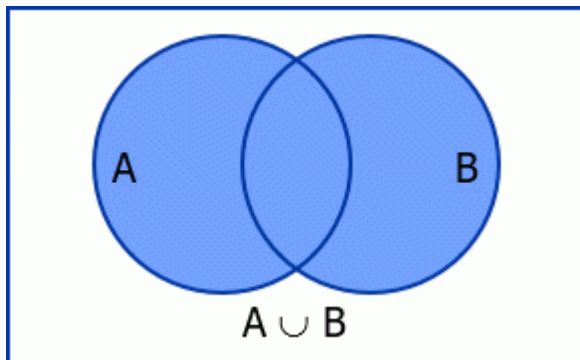**Lab 01**
Title: To find the union of two numbers

Given two sets A and B, the union is the set that contains all elements or objects that belong to A and to B.

For example:

A = { **b**, 1, 2, **4**, 6 } and B = { **4**, a, **b**, c, d, f }

A ∩ B = { 4, b }

$A \cup B$

Program:
```
#include<stdio.h>
#include<conio.h>
int main()
{
  int a[10],b[10],i,c[10],j,k=0,n1,n2;
  // taking input set A
  printf("Enter number of  element of set A\n");
  scanf("%d",&n1);
  printf("Enter the element of set A \n");
  for(i=0;i<n1;i++)
    scanf("%d",&a[i]);
   // taking input set B
  printf("Enter number of element of set B\n");
  scanf("%d",&n2);
  printf("Enter the element of set B \n");
  for(i=0;i<n2;i++)
    scanf("%d",&b[i]);
  // logic for calculate union

  // copy the element of set A in set C
  for(i=0;i<n1;i++)
  {
      // repeted element is not allowed so we check is any value repeted
```

```c
    for(j=0;j<k;j++)
    {
        if(c[j]==a[i])
            break;
    }
    if(j==k) //if not repesated then store value in set c
    {
        c[k]=a[i];
        k++;
    }
}
// copy element of set B in set C
for(i=0;i<n2;i++)
{
    // check for repeted element
    for(j=0;j<k;j++)
    {
        if(c[j]==b[i])
            break;
    }
    if(j==k) // if element is not repeted then store in set C
    {
        c[k]=b[i];
        k++;
    }
}

// printing of union of set A and set B
printf("Union of set A and B is:-\n");
for(i=0;i<k;i++)
    printf("%d ",c[i]);
}
```

```
sulavparajuli@pop-os:/opt/lampp/htdocs/admintmp/discrete$ gcc lab.cpp
sulavparajuli@pop-os:/opt/lampp/htdocs/admintmp/discrete$ ./a.out
Enter number of  element of set A
2
Enter the element of set A
1 3
Enter number of element of set B
3
Enter the element of set B
2 5 4
Union of set A and B is:-
1 3 2 5 4 sulavparajuli@pop-os:/opt/lampp/htdocs/admintmp/discrete$ []
```
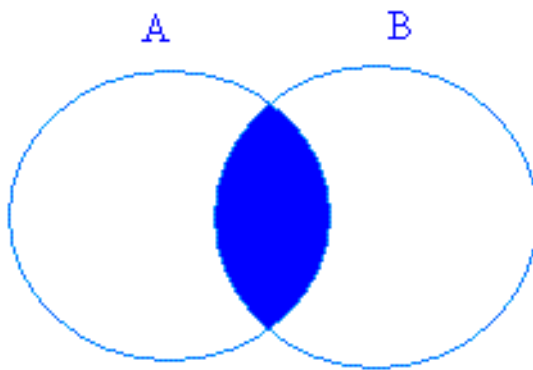
Lab 02

Title: To find the intersection of two numbers

Given two sets A and B, the intersection is the set that contains elements or objects that belong to A and to B at the same time.

For example:

A = { **b**, 1, 2, **4**, 6 } and B = { **4**, a, **b**, c, d, f }

A ∩ B = { 4, b }

Shaded region is the intersection of A and B

Program:

```
#include<stdio.h>
int main()
{
    int a[100],b[100],c[100],n1,n2,n,k=0,i,j;

    // taking input of set A

    printf("Enter number of element of set A\n");
    scanf("%d",&n1);
    printf("Enter elements of set A\n");
    for(i=0;i<n1;i++)
      scanf("%d",&a[i]);

    // taking input set B

    printf("Enter number of element of set B\n");
    scanf("%d",&n2);
```

```c
    printf("Enter elements of set B\n");
    for( i=0;i<n2;i++)
        scanf("%d",&b[i]);

    // Logic for intersection

    for( i=0;i<n1;i++)
    {
        for(j=0;j<n2;j++)
        {
            if(a[i]==b[j])
            {
                c[k]=a[i];
                k++;
            }
        }

    }

    // Printing the elements of intersection of set A and set B
    printf("intersection of set A and set B is:-\n");
    for(i=0;i<k;i++)
        printf("%d ",c[i]);

    return 0;
}
```

OUTPUT

```
sulavparajuli@pop-os:/opt/lampp/htdocs/admintmp/discrete$ gcc lab.cpp
sulavparajuli@pop-os:/opt/lampp/htdocs/admintmp/discrete$ ./a.out
Enter number of element of set A
2
Enter elements of set A
2 3
Enter number of element of set B
3
Enter elements of set B
3 5 6
intersection of set A and set B is:-
3
```