AIM: Write a program in C to create two sets and perform the Union and intersection operation on sets.

THEORY: A set is a collection of well defined distinct objects, considered as an object in its own right. For example, the numbers 2, 4, and 6 are distinct objects when considered separately, but when they are considered collectively they form a single set of size three, written {2, 4, 6}. The concept of a set is one of the most fundamental in mathematics.

Some basic properties of unions:

- $A \cup B = B \cup A$.
- $A \cup (B \cup C) = (A \cup B) \cup C$.
- $A \subseteq (A \cup B)$.
- $A \cup A = A$.
- $A \cup \emptyset = A$.
- $A \subseteq B$ if and only if $A \cup B = B$.

Some basic properties of intersections:

- $A \cap B = B \cap A$.
- $A \cap (B \cap C) = (A \cap B) \cap C$.
- $A \cap B \subseteq A$.
- $A \cap A = A$.
- $A \cap \emptyset = \emptyset$.
- $A \subseteq B$ if and only if $A \cap B = A$.

IMPLEMENTATION:

```
#include<stdio.h>

void Union(int set1[10], int set2[10], int m, int n);

void Intersection(int set1[10], int set2[10], int m, int n);

void main()
```

```
{
int a[10], b[10], m, n, i, j;
   int choice;
   printf("\nEnter the number of elements in first set:\t");
   scanf("%d",&m);
   printf("\nEnter the elements:\n");
   for(i=0;i<m;i++)
      {
         scanf("%d",&a[i]);
      }
   printf("\nElement of First set:\n");
   for(i=0;i<m;i++)
      {
         printf("%d\t",a[i]);
      }
   printf("\nEnter the number of elements in second set:\t");
   scanf("%d",&n);
   printf("\nEnter the elements:\n");
   for(i=0;i<n;i++)
      {
         scanf("%d",&b[i]);
      }
   printf("\nElement of second set:\n");
   for(i=0;i<n;i++)
```

```
        {
            printf("%d\t",b[i]);
        }
    for(;;)
        {

printf("\nMenu\n");
printf("_____");
            printf("\n1.Union\n2.Intersection\n");
printf("3.exit\n");
            printf("\nEnter your choice:\t");
            scanf("%d",&choice);
            switch(choice)
             {
                case 1: Union(a,b,m,n); break;
                case 2: Intersection(a,b,m,n); break;
                case 3: exit(0);
                default: printf("\nInvalid choice\n");
            }
        }
}

void Union(int a[10], int b[10], int m, int n)
{
    int c[20], i, j, k=0, flag=0;
    for(i=0;i<m;i++)
```

```
        {
           c[k]=a[i];
           k++;
        }
     for(i=0;i<n;i++)
        {
           flag=0;
           for(j=0;j<m;j++)
           {

 if(b[i]==c[j])
              {
                 flag=1;
break;
              }
           }
           if(flag==0)
              {
                 c[k]=b[i];
                 k++;
              }
        }
     printf("\nElement of resultant set:\n");
     for(i=0;i<k;i++)
        {
           printf("\t%d\t",c[i]);
```

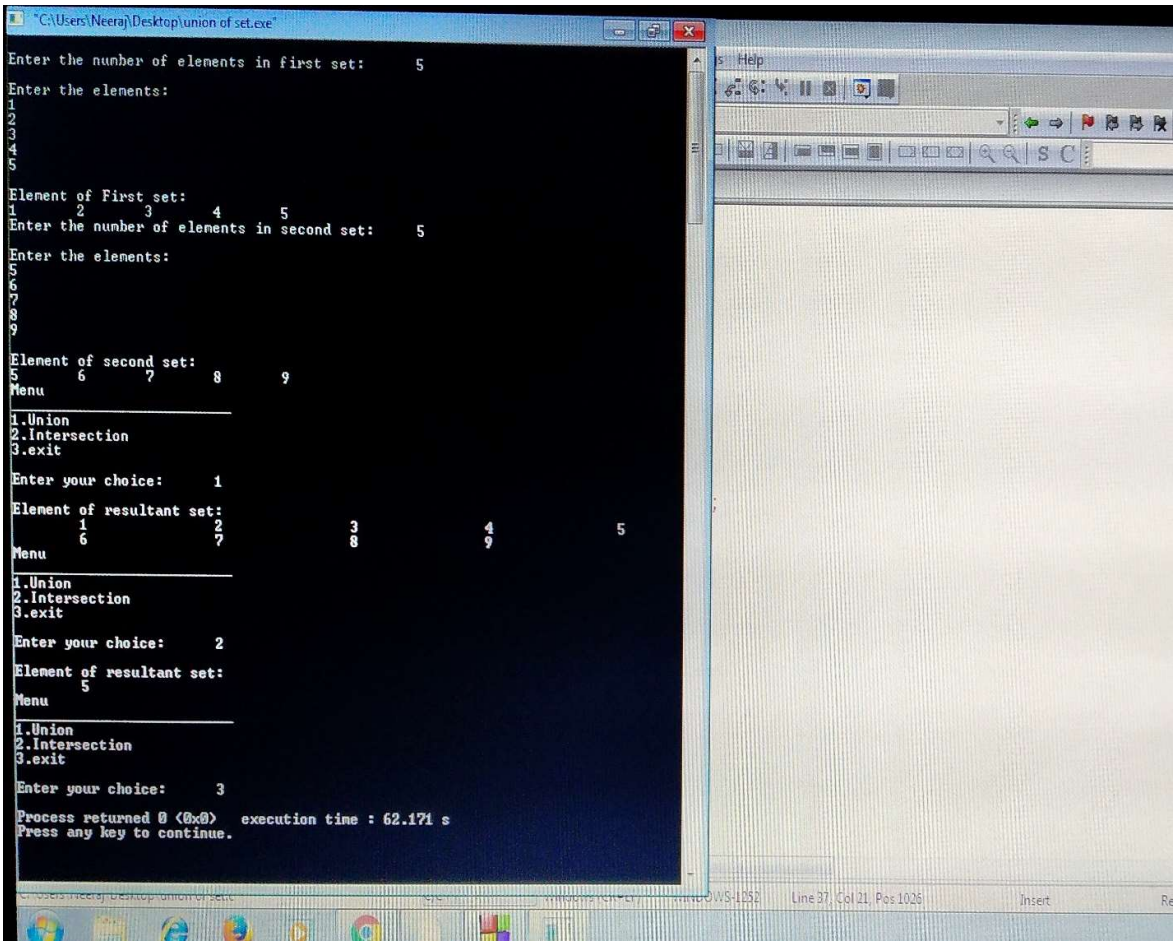
```
        }
    }

    void Intersection(int a[10], int b[10], int m, int n)
    {
       int c[20], i, j, k=0, flag=0;
       for(i=0;i<m;i++)
          {
             flag=0;
             for(j=0;j<n;j++)
             {
                if(a[i]==b[j])
                {

    flag=1;
    break;
                }
             }
    if(flag==1)
                {
                   c[k]=a[i];
                   k++;
                }
          }
       if(k==0)
          {
```

```
            printf("\nResultant set is null set!\n");
        }
    else
        {
            printf("\nElement of resultant set:\n");
            for(i=0;i<k;i++)
            {
                printf("\t%d\t",c[i]);
            }
        }
}
```

OUTPUT:


"C:\Users\Neeraj\Desktop\union of set.exe"
Enter the number of elements in first set: 5
Enter the elements:
1
2
3
4
5
Element of First set:
1 2 3 4 5
Enter the number of elements in second set: 5
Enter the elements:
5
6
7
8
9
Element of second set:
5 6 7 8 9
Menu
1.Union
2.Intersection
3.exit
Enter your choice: 1
Element of resultant set:
1 2 3 4 5
6 7 8 9
Menu
1.Union
2.Intersection
3.exit
Enter your choice: 2
Element of resultant set:
5
Menu
1.Union
2.Intersection
3.exit
Enter your choice: 3
Process returned 0 (0x0) execution time : 62.171 s
Press any key to continue.
Help
Line 37, Col 21, Pos 1026
Insert