DATA STRUCTURES

Roll No: _____

Name: _____

INDEX

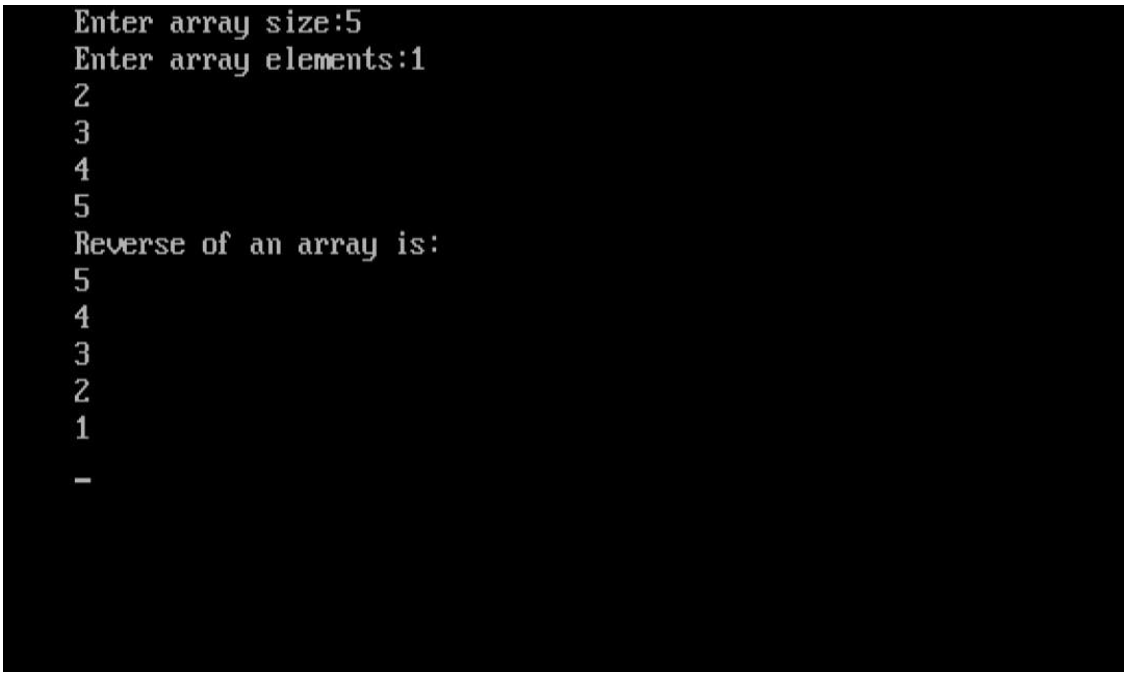
PRACTIAL NO.1 (A)

AIM: Write a program to store the element in 1-d array and perform the operation **REVERSING** the elements.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int arr[20],size,i,j,temp;
   cout<<"Enter array size:";
   cin>>size;
   cout<<"Enter array elements:";
   for(i=0;i<size;i++)
   {
        cin>>arr[i];
   }
   j=i-1;
   i=0;
   while(i<j)
   {
        temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;
        i++;
        j--;
   }
   cout<<"Reverse of an array is:\n";
   for(i=0;i<size;i++)
   {
   cout<<arr[i]<<"\n";
   }
   getch();
}
```

OUTPUT:-


Enter array size:5
Enter array elements:1
2
3
4
5
Reverse of an array is:
5
4
3
2
1

PRACTIAL NO.1 (B)

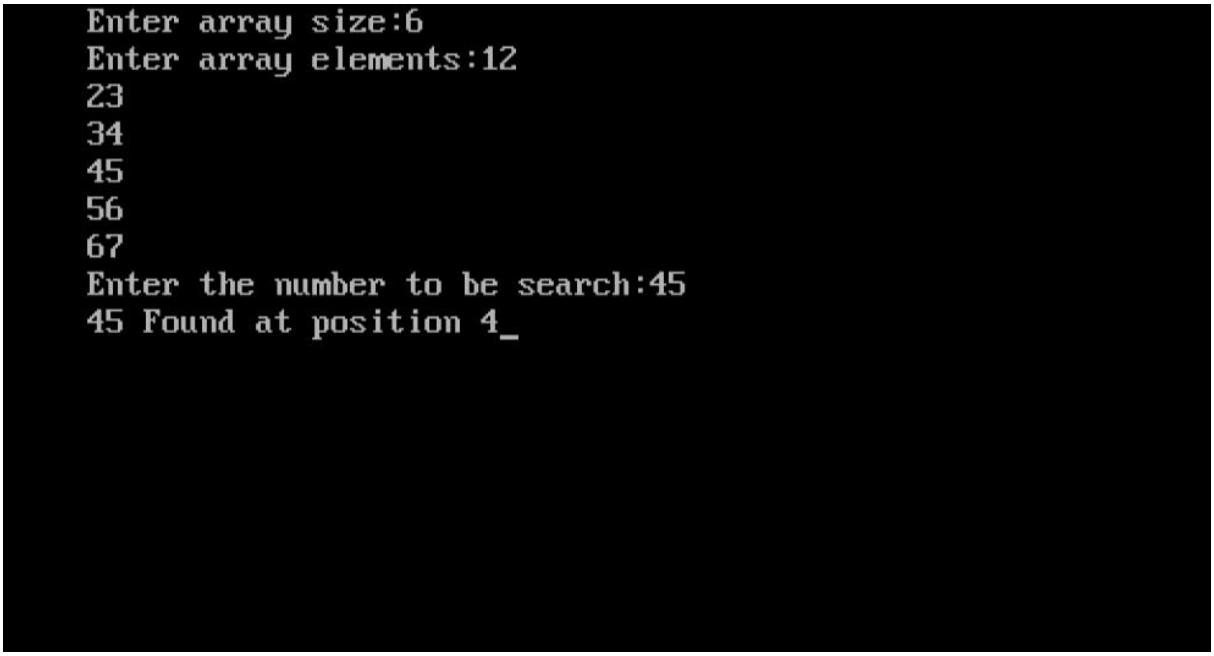
AIM: Write a program to store the elements in 1-d array and perform the operation **LINEAR SEARCHING** of the elements.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int arr[10],i,num,n,c=0,pos;
   cout<<"Enter array size:";
   cin>>n;
   cout<<"Enter array elements:";
   for(i=0;i<n;i++)
   {
        cin>>arr[i];
   }
   cout<<"Enter the number to be search:";
   cin>>num;
   for(i=0;i<n;i++)
   {
        if(arr[i]==num)
        {
           c=1;
           pos=i+1;
           break;
        }
   }
   if(c==0)
   {
        cout<<"Number not found!";
   }
   else
   {
```

```
        cout<<num<<"\n Found at position "<<pos;
    }
    getch();
}
```

OUTPUT:-



PRACTIAL NO.2 (A)

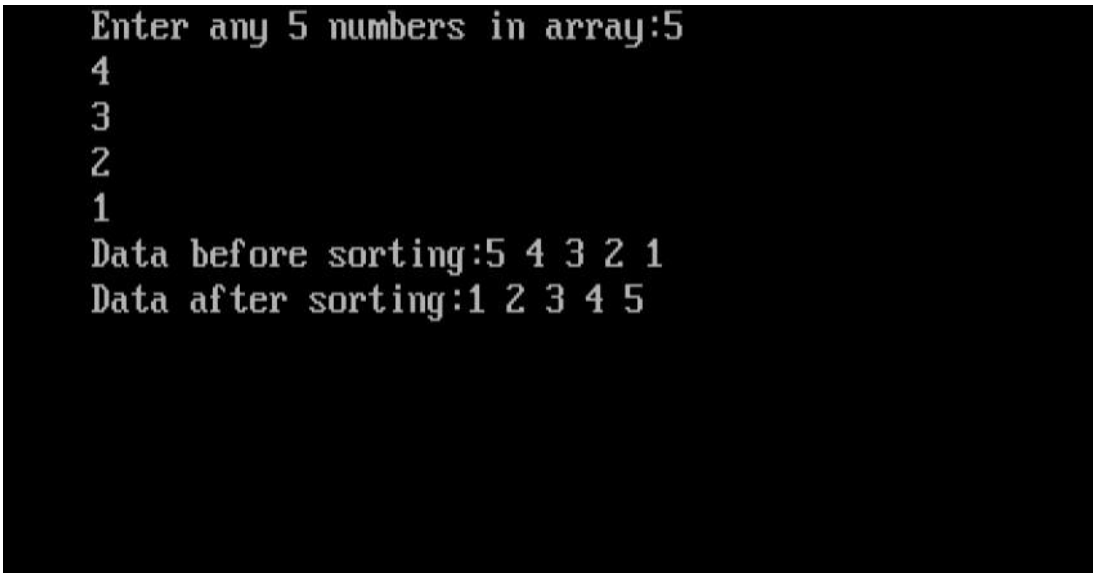
AIM: Write a program to store the elements in 1-d array and perform the operation **SORTING IN ASCENDING ORDER**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int i,a[10],temp,j;
   cout<<"Enter any 5 numbers in array:";
   for(i=0;i<=4;i++)
   {
        cin>>a[i];
   }
   cout<<"Data before sorting:";
   for(j=0;j<5;j++)
   {
        cout<<a[j]<<" ";
   }
   for(i=0;i<=4;i++)
   {
        for(j=0;j<=4-i;j++)
        {
           if(a[j]>a[j+1])
           {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
           }
        }
   }
   cout<<"\nData after sorting:";
   for(j=0;j<5;j++)
   {
```

```
        cout<<a[j]<<" ";
    }
    getch();
}
```

OUTPUT:-



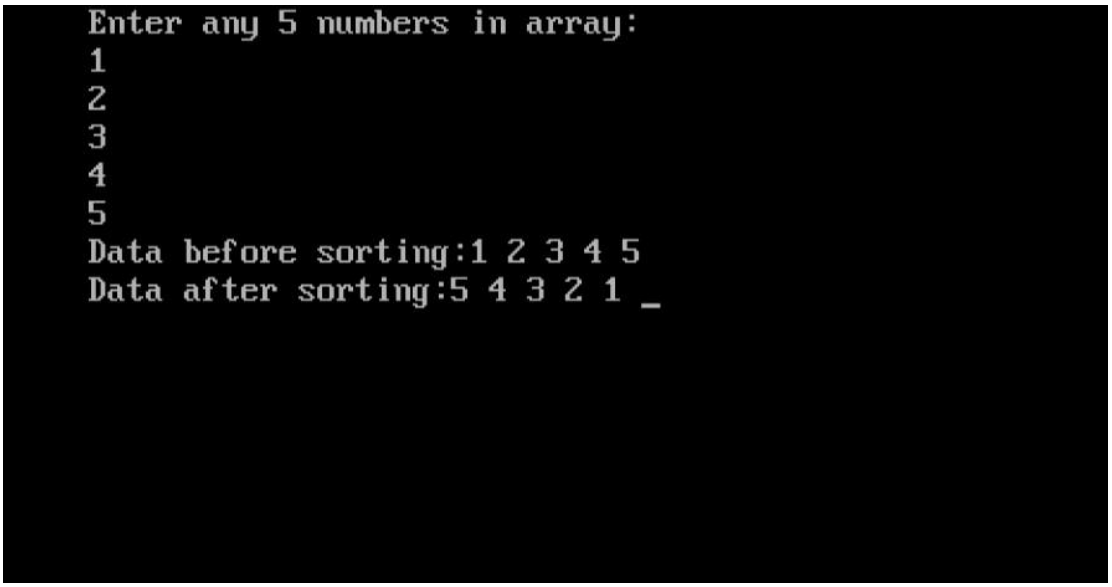
PRACTIAL NO.2 (B)

AIM: Write a program to store the elements in 1-d array and perform the operation **SORTING IN DESCENDING ORDER**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int i,a[10],temp,j;
   cout<<"Enter any 5 numbers in array:\n";
   for(i=0;i<=4;i++)
   {
         cin>>a[i];
   }
   cout<<"Data before sorting:";
   for(j=0;j<5;j++)
   {
         cout<<a[j]<<" ";
   }
   for(i=0;i<=4;i++)
   {
         for(j=0;j<=4-i;j++)
         {
            if(a[j]>a[j+1])
            {
                  temp=a[j];
                  a[j]=a[j+1];
                  a[j+1]=temp;
            }
         }
   }
   cout<<"\nData after sorting:";
   for(j=4;j>=0;j--)
   {
         cout<<a[j]<<" ";
   }
   getch();
}
```

OUTPUT:-


Enter any 5 numbers in array:
1
2
3
4
5
Data before sorting:1 2 3 4 5
Data after sorting:5 4 3 2 1

PRACTIAL NO.2 (C)

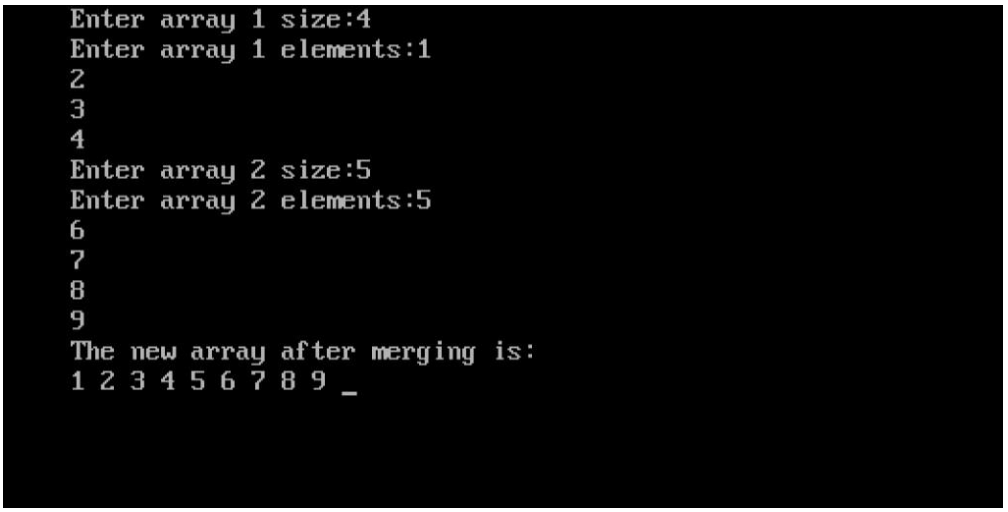
AIM: Read the two arrays from the user and **MERGE** them and display the elements in **SORTED ORDER**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int arr1[10],arr2[10],size1,size2,size,i,j,k,merge[20];
   cout<<"Enter array 1 size:";
   cin>>size1;
   cout<<"Enter array 1 elements:";
   for(i=0;i<size1;i++)
   {
        cin>>arr1[i];
   }
   cout<<"Enter array 2 size:";
   cin>>size2;
   cout<<"Enter array 2 elements:";
   for(i=0;i<size2;i++)
   {
        cin>>arr2[i];
   }
   for(i=0;i<size1;i++)
   {
       merge[i]=arr1[i];
   }
   size=size1+size2;
   for(i=0,k=size1;k<size&&i<size2;i++,k++)
   {
         merge[k]=arr2[i];
   }
   cout<<"The new array after merging is:\n";
   for(i=0;i<size;i++)
   {
```

```
        cout<<merge[i]<<" ";
    }
    getch();
}
```

OUTPUT:-



PRACTIAL NO.3 (A)

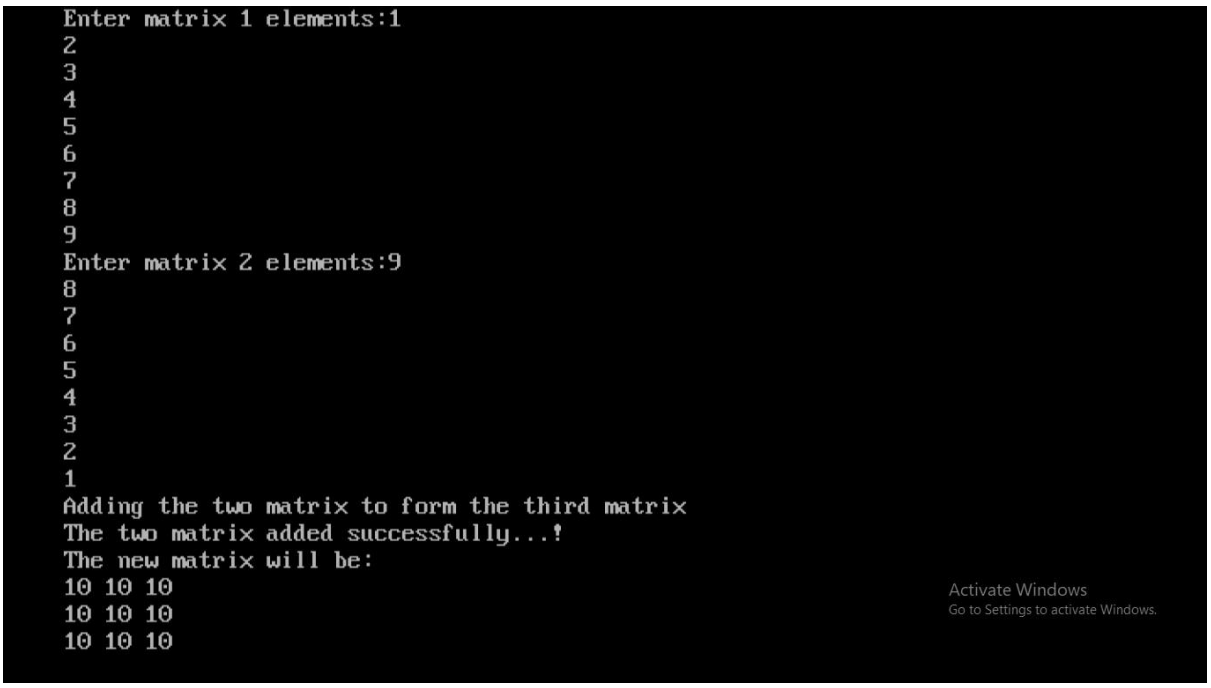
AIM: Write a program to perform the matrix **ADDITION** operation.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int mat1[3][3],mat2[3][3],i,j,mat3[3][3];
   cout<<"Enter matrix 1 elements:";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat1[i][j];
        }
   }
   cout<<"Enter matrix 2 elements:";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat2[i][j];
        }
   }
   cout<<"Adding the two matrix to form the third matrix \n";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
        mat3[i][j]=mat1[i][j]+mat2[i][j];
        }
   }
   cout<<"The two matrix added successfully...!";
   cout<<"\nThe new matrix will be:\n";
   for(i=0;i<3;i++)
```

```
    {
        for(j=0;j<3;j++)
        {
           cout<<mat3[i][j]<<" ";
        }
        cout<<"\n";
    }
    getch();
}
```

OUTPUT :-



PRACTIAL NO.3 (B)

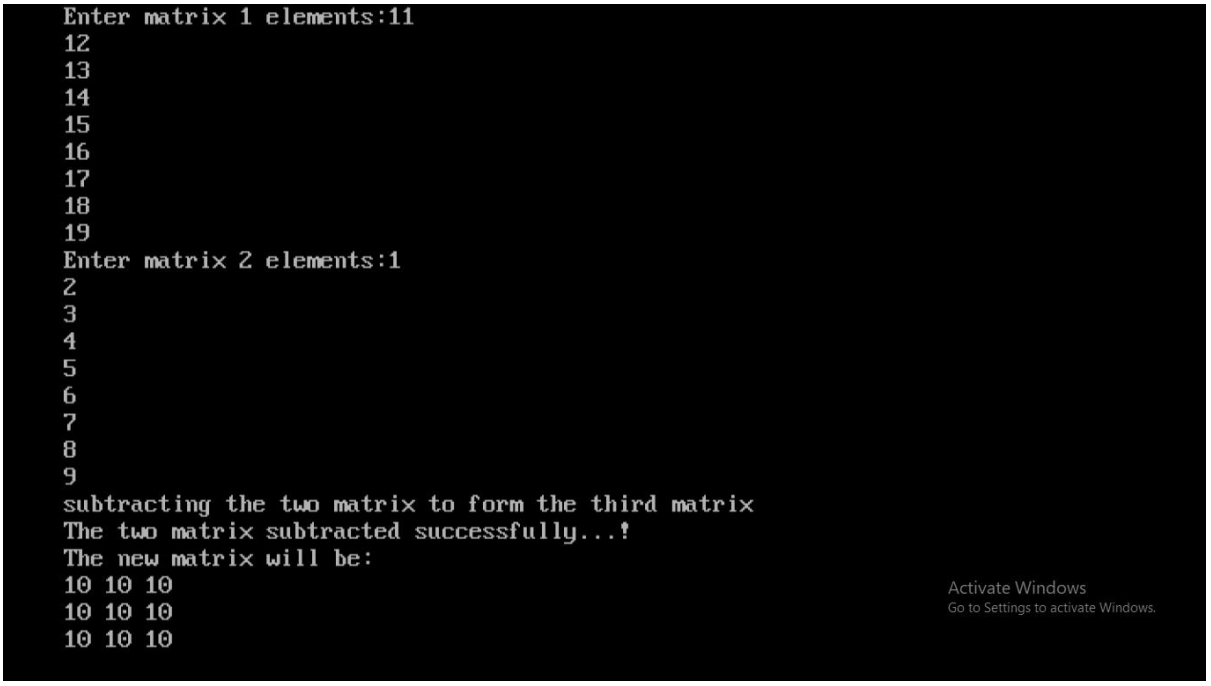
AIM: Write a program to perform the matrix **SUBTRACTING** operation.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int mat1[3][3],mat2[3][3],i,j,mat3[3][3];
   cout<<"Enter matrix 1 elements:";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat1[i][j];
        }
   }
   cout<<"Enter matrix 2 elements:";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat2[i][j];
        }
   }
   cout<<"subtracting the two matrix to form the third matrix \n";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
        mat3[i][j]=mat1[i][j]-mat2[i][j];
        }
   }
   cout<<"The two matrix subtracted successfully...!";
   cout<<"\nThe new matrix will be:\n";
   for(i=0;i<3;i++)
```

```
    {
        for(j=0;j<3;j++)
        {
           cout<<mat3[i][j]<<" ";
        }
        cout<<"\n";
    }
    getch();
}
```

OUTPUT:-



PRACTIAL NO.3 (C)

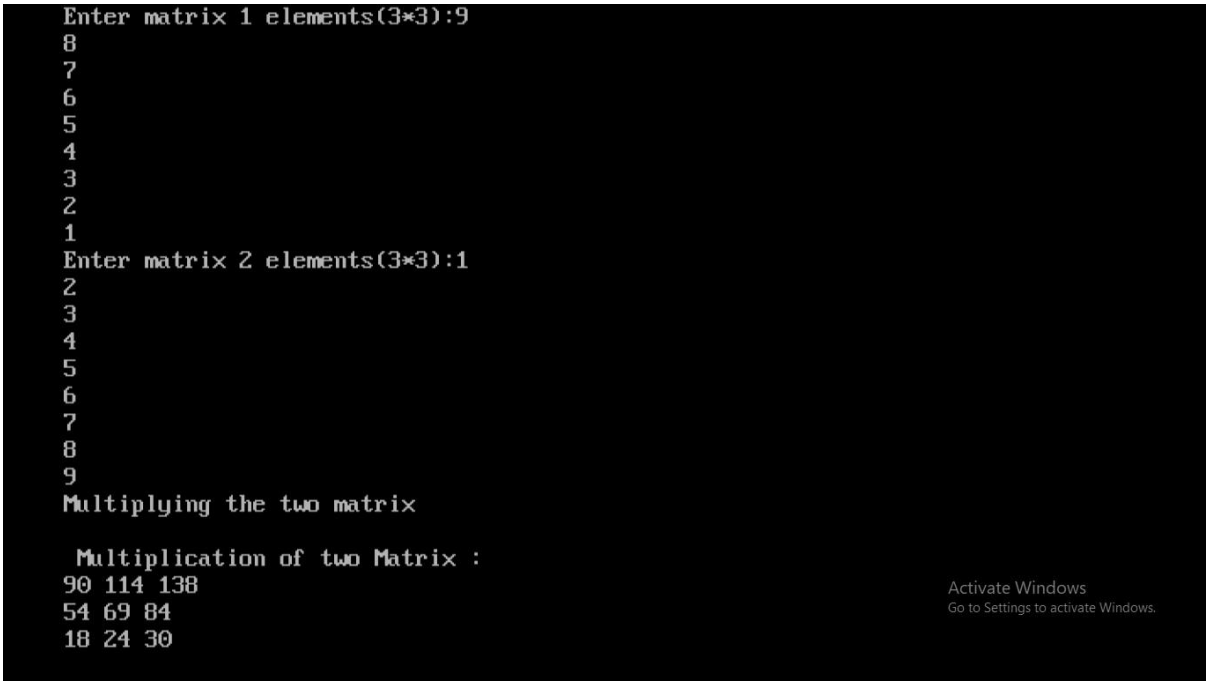
AIM: Write a program to perform the matrix **MULTIPLICATION** operation.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int mat1[3][3],mat2[3][3],mat3[3][3],sum=0,i,j,k;
   cout<<"Enter matrix 1 elements(3*3):";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat1[i][j];
        }
   }
   cout<<"Enter matrix 2 elements(3*3):";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>mat2[i][j];
        }
   }
   cout<<"Multiplying the two matrix \n";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           sum=0;
           for(k=0;k<3;k++)
           {
                sum=sum+mat1[i][k] * mat2[k][j];
           }
           mat3[i][j]=sum;
```

```
        }
    }
    cout<<"\n Multiplication of two Matrix \n:";
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
           cout<<mat3[i][j]<<" ";
        }
        cout<<"\n";
    }
    getch()
}
```

OUTPUT:-



PRACTIAL NO.3 (D)

AIM: Write a program to perform the matrix **TRANSPOSE** operation.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int arr[3][3],i,j;
   cout<<"Enter the 3*3 Array Elements:";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cin>>arr[i][j];
        }
   }
   cout<<"Transpose of the marix is:\n";
   for(i=0;i<3;i++)
   {
        for(j=0;j<3;j++)
        {
           cout<<arr[j][i]<<" ";
        }
        cout<<"\n";
   }
   getch();
}
```

OUTPUT:-

```
Enter the 3*3 Array Elements:1
2
3
4
5
6
7
8
9
Transpose of the marix is:
1 4 7
2 5 8
3 6 9
```

PRACTIAL NO.4 (A)

AIM: Write a program to implement the **CONCEPT OF STACK** with push, pop, display and exit operations.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class stack
{
   int stk[5];
   int top;
   public:
   stack()
   {
        top=-1;
   }
   void push(int x)
   {
        if(top>4)
        {
        cout<<"stack over flow";
        return;
        }
        stk[++top]=x;
        cout<<"inserted"<<x;
   }
   void pop()
   {
        if(top<0)
        {
           cout<<"stack under flow";
           return;
        }
        cout<<"deleted"<<stk[top--];
   }
   void display()
```

```
    {
        if(top<0)
        {
            cout<<"stack empty" ;
            return;
        }
        for(int i=top;i>=0;i--)
        {
            cout<<stk[i]<<" ";
        }
    }
};
void main()
{
    clrscr();
    int ch;
    stack st;
    while(1)
    {
        cout<<"\n 1.push 2.pop 3.display 4.exit\n enter your choice ";
        cin>>ch;
        switch(ch)
        {
            case 1:cout<<"enter the element ";
            cin>>ch;
            st.push(ch);
            break;
            case 2:st.pop();
            break;
            case 3:st.display();
            break;
            case 4:exit(0);
            default:
            cout<<”Invalid Input”;
            break;
        }
    }
    getch();
```

```
}
```

OUTPUT:-

```
 1.push 2.pop 3.display 4.exit
 enter your choice 1
enter the element 1
inserted 1
 1.push 2.pop 3.display 4.exit
 enter your choice 1
enter the element 2
inserted 2
 1.push 2.pop 3.display 4.exit
 enter your choice 1
enter the element 3
inserted 3
 1.push 2.pop 3.display 4.exit
 enter your choice 3
3 2 1
 1.push 2.pop 3.display 4.exit
 enter your choice 2
deleted 3
 1.push 2.pop 3.display 4.exit
 enter your choice 2
deleted 2
 1.push 2.pop 3.display 4.exit
 enter your choice
```

PRACTIAL NO.5 (A)

AIM: Write a program to implement the **CONCEPT OF QUEUE** with insert, delete, display and exit operations.

SOURCE CODE:-

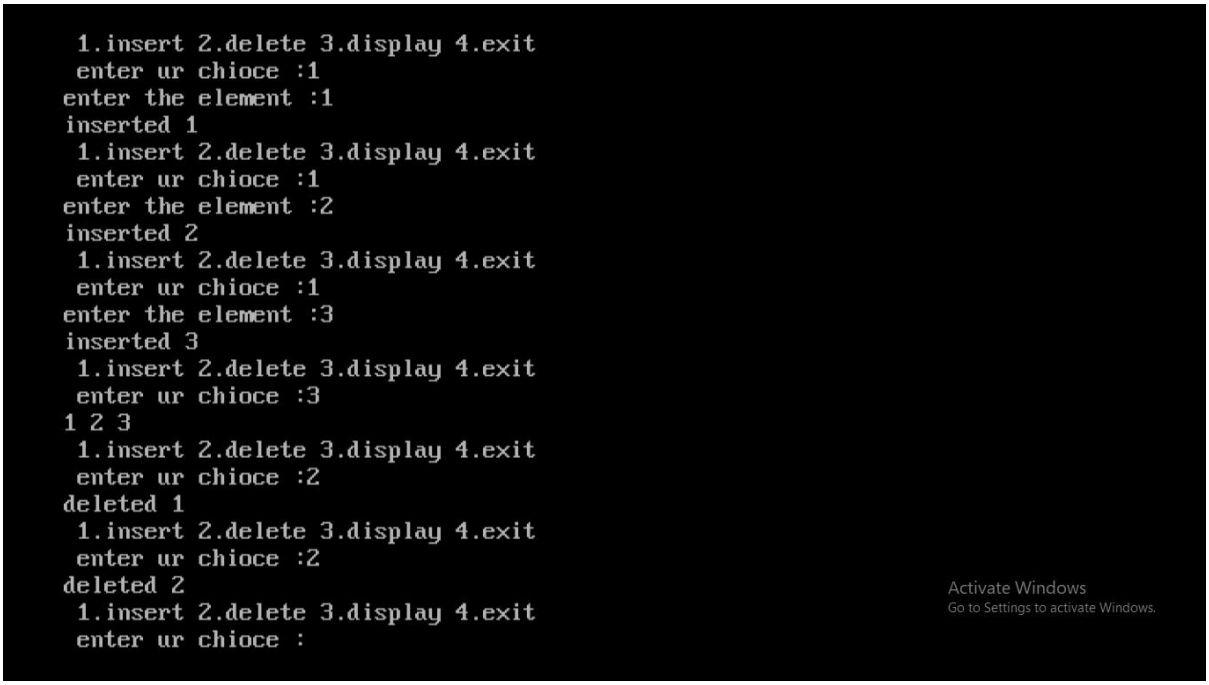
```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class queue
{
   int queue1[5];
   int rear,front;
   public:
   queue()
   {
        rear=-1;
        front=-1;
   }
   void insert(int x)
   {
        if(rear>4)
        {
        cout<<"queue overflow";
        return;
        }
        queue1[++rear]=x;
        cout<<"inserted"<<x;
   }
   void delet()
   {
        if(front==rear)
        {
           cout<<"queue under flow";
           return;
        }
        cout<<"deleted"<<queue1[++front];
```

```
    }
    void display()
    {
        if(rear==front)
        {
            cout<<"queue empty";
            return;
        }
        for(int i=front+1;i<=rear;i++)
        {
            cout<<queue1[i]<<" ";
        }
    }
};
void main()
{
    clrscr();
    int ch;
    queue qu;
    while(1)
    {
        cout<<"\n 1.insert 2.delet 3.display 4.exit\n enter ur chioce :";
        cin>>ch;
        switch(ch)
        {
            case 1:cout<<"enter the element :";
            cin>>ch;
            qu.insert(ch);
            break;
            case 2:qu.delet();
            break;
            case 3:qu.display();
            break;
            case 4:exit(0);
            default :
            cout<<”Invalid Input”;
            break;
        }
```

```
}
getch();
```

}

OUTPUT:-


1.insert 2.delete 3.display 4.exit
enter ur chioce :1
enter the element :1
inserted 1
1.insert 2.delete 3.display 4.exit
enter ur chioce :1
enter the element :2
inserted 2
1.insert 2.delete 3.display 4.exit
enter ur chioce :1
enter the element :3
inserted 3
1.insert 2.delete 3.display 4.exit
enter ur chioce :3
1 2 3
1.insert 2.delete 3.display 4.exit
enter ur chioce :2
deleted 1
1.insert 2.delete 3.display 4.exit
enter ur chioce :2
deleted 2
1.insert 2.delete 3.display 4.exit
enter ur chioce :
Activate Windows
Go to Settings to activate Windows.

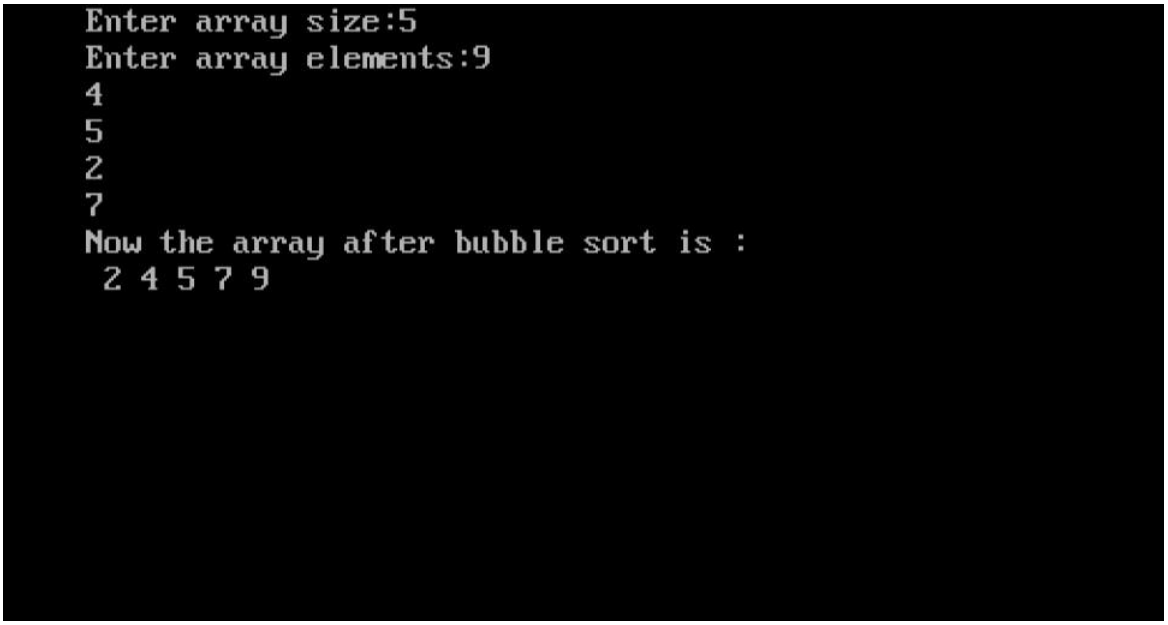
PRACTIAL NO.6 (A)

AIM: Write a program to implement **BUBBLE SORT**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int size,arr[10],i,j,temp;
   cout<<"Enter array size:";
   cin>>size;
   cout<<"Enter array elements:";
   for(i=0;i<size;++i)
   {
        cin>>arr[i];
   }
   for(i=1;i<size;++i)
   {
        for(j=0;j<(size-i);++j)
        {
           if(arr[j]>arr[j+1])
           {
                temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
           }
        }
   }
   cout<<"Now the array after bubble sort is :\n";
   for(i=0;i<size;++i)
   {
        cout<<" "<<arr[i];
   }
   getch();
}
```

OUTPUT:-


Enter array size:5
Enter array elements:9
4
5
2
7
Now the array after bubble sort is :
2 4 5 7 9

PRACTIAL NO.6 (B)

AIM: Write a program to implement **SELECTION SORT**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int size,arr[10],i,j,temp;
   cout<<"Enter array size:";
   cin>>size;
   cout<<"Enter array elements:";
   for(i=0;i<size;i++)
   {
        cin>>arr[i];
   }
   for(i=0;i<size;i++)
   {
        for(j=i+1;j<size;j++)
        {
           if(arr[i]>arr[j])
           {
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
           }
        }
   }
   cout<<"Now the array after sorting is :\n";
   for(i=0;i<size;i++)
   {
        cout<<arr[i]<<" ";
   }
   getch();
}
```

OUTPUT:-

```
Enter array size:5
Enter array elements:9
3
6
7
2
Now the array after sorting is :
2 3 6 7 9
```

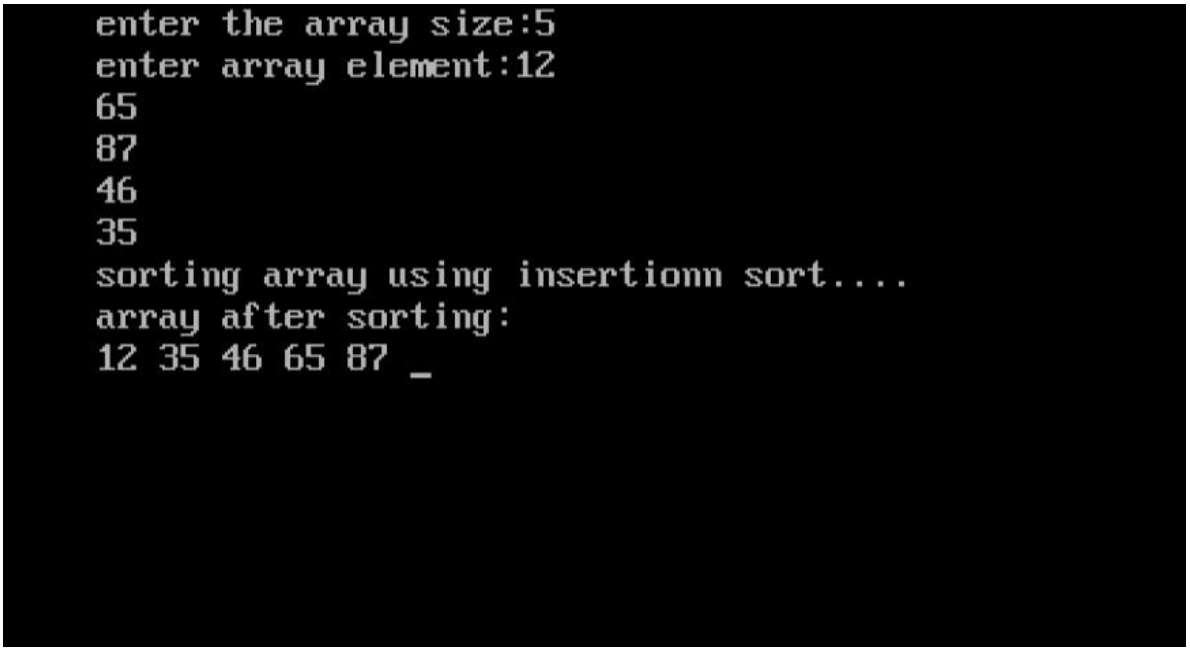
PRACTIAL NO.7 (A)

AIM: Write a program to implement **INSERTION SORT**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int size,arr[50],i,j,temp;
   cout<<"enter the array size:";
   cin>>size;
   cout<<"enter array element:";
   for(i=0;i<size;i++)
   {
        cin>>arr[i];
   }
   cout<<"sorting array using insertionn sort....\n";
   for(i=1;i<size;i++)
   {
        temp=arr[i];
        j=i-1;
        while((temp<arr[j])&&(j>=0))
        {
           arr[j+1]=arr[j];
           j=j-1;
        }
        arr[j+1]=temp;
   }
   cout<<"array after sorting: \n";
   for(i=0;i<size;i++)
   {
        cout<<arr[i]<<" ";
   }
   getch();
}
```

OUTPUT:-


enter the array size:5
enter array element:12
65
87
46
35
sorting array using insertionn sort....
array after sorting:
12 35 46 65 87 _

PRACTIAL NO.8 (A)

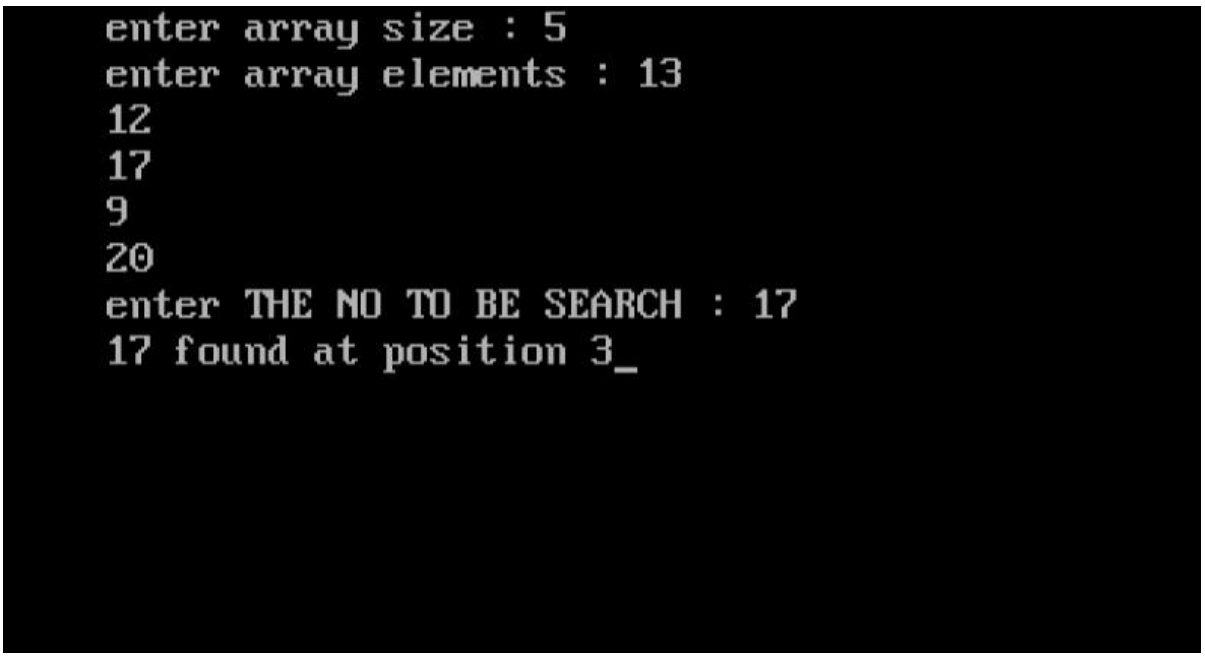
AIM: Write a program to implement **SEQUENTIAL SEARCH**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int arr[50],size,i,j=0,temp,pos;
   cout<<"enter array size : ";
   cin>>size;
   cout<<"enter array elements : ";
   for(i=0;i<size;i++)
   {
        cin>>arr[i];
   }
   cout<<"enter THE NO TO BE SEARCH : ";
   cin>>temp;
   for(i=0;i<size;i++)
   {
        if(arr[i]==temp)
        {
           j=1;
           pos=i+1;
           break;
        }
   }
   if(j==0)
   {
        cout<<" no. not found";
   }
   else
   {
        cout<<temp<<" found at position "<<pos;
   }
```

```
    getch();
}
```

OUTPUT:-



PRACTIAL NO.8 (B)

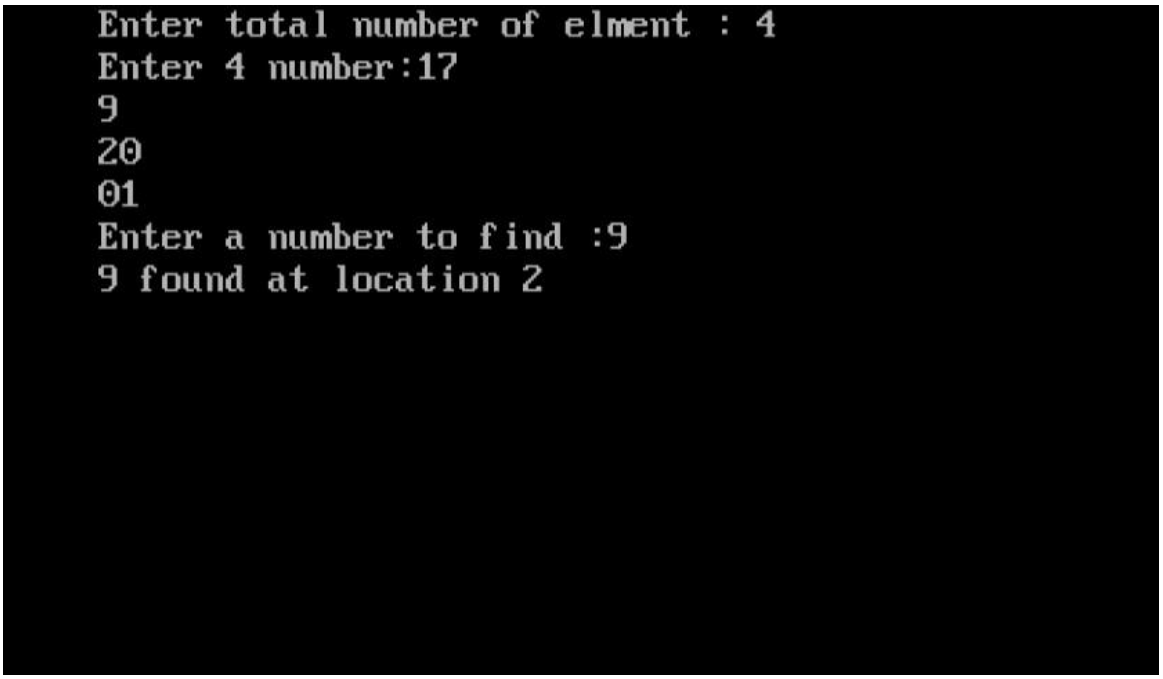
AIM: Write a program to search the element using **BINARY SEARCH**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void main()
{
   clrscr();
   int n,i,arr[50],search,first,last,middle;
   cout<<"Enter total number of elment : ";
   cin>>n;
   cout<<"Enter "<<n<<" number:";
   for(i=0;i<n;i++)
   {
        cin>>arr[i];
   }
   cout<<"Enter a number to find :";
   cin>>search;
   first=0;
   last=n-1;
   middle=(first+last)/2;
   while(first<=last)
   {
        if(arr[middle]<search)
        {
           first=middle+1;
        }
        else if(arr[middle]==search)
        {
           cout<<search<<" found at location "<<middle+1<<"\n";
           break;
        }
        else
        {
           last=middle-1;
```

```
        }
        middle=(first+last)/2;
    }
    if(first>last)
    {
        cout<<" not found "<<search<<" is not prsent in the list";
    }
    getch();
}
```

OUTPUT:-



PRACTIAL NO.9 (A)

AIM: Write a program to implement **TOWER OF HANOI PROBLEM**.

SOURCE CODE:-

```
#include<iostream.h>
#include<conio.h>
void tower(int,char,char,char);
int main()
{
   clrscr();
   int num;
   cout<<"enter the number of disks:";
   cin>>num;
   cout<<"the squence of moves involvedin the towers of the hanoi are:\n";
   tower(num,'A','C','B');
   return 0;
}
void tower(int num,char frompeg,char topeg,char auxpeg)
{
   if(num==1)
   {
        cout<<"\n move disk 1 from peg "<<frompeg<<" to peg "<<topeg;
        return;
   }
   tower(num-1,frompeg,auxpeg,topeg);
   cout<<"\n move disk "<<num<<" frompeg "<<from peg<<" to peg "<<topeg;
   tower(num-1,auxpeg,topeg,frompeg);
   getch();
}
```

OUTPUT:-

```
enter the number of disks:3
the squence of moves involvedin the towers of the hanoi are:

 move disk 1 from peg A to peg C
 move disk 2 from peg A to peg B
 move disk 1 from peg C to peg B
 move disk 3 from peg A to peg C
 move disk 1 from peg B to peg A
 move disk 2 from peg B to peg C
 move disk 1 from peg A to peg C_
```