ABHINAV NARAYANAN
3SU21CC007
LAB ON DS

INDEX

PART-A

PART-B

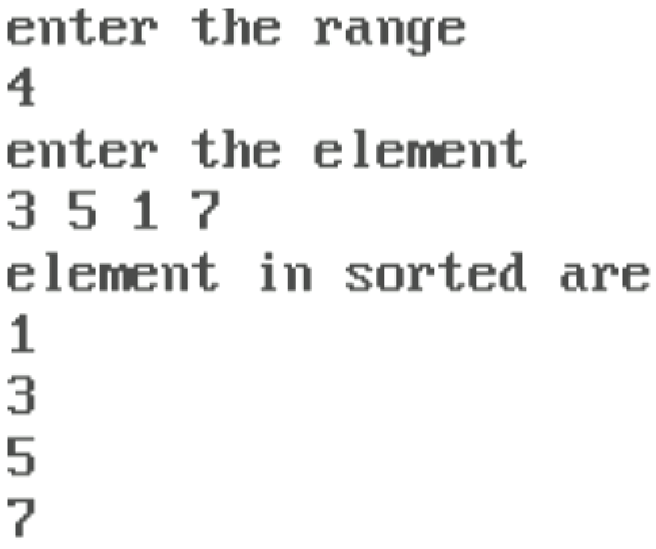
ABHINAV NARAYANAN
3SU21CC007
LAB ON DS

PROGRAM-1

Q: WRITE A C++ PROGRAM TO DEMONSTRATE BUBBLE SORT.

```
#include<conio.h>
#include<iomanip.h>
int i,temp,j,n,a[15];
class bubble
{
        public:
                void getdata();
                void sort();
                void display();
};
        void bubble::getdata()
{
        cout<<"enter the range"<<endl;
        cin>>n;
        cout<<"enter the element"<<endl;
        for(i=0;i<n;i++)
        cin>>a[i];
}
        void bubble::sort()
{
        int i,j,temp;
        for(i=0;i<n-1;i++)
{
        for(j=0;j<n-i-1;j++)
        if(a[j]>a[j+1])
{
        temp=a[j];
        a[j]=a[j+1];
        a[j+1]=temp;
}
}
}
        void bubble::display()
{
        cout<<"element in sorted are"<<endl;
        for(i=0;i<n;i++)
        cout<<a[i]<<endl;
}
  void main()
{
  bubble s;
  clrscr();
  s.getdata();
  s.sort();
  s.display();
  getch();
}
```

OUTPUT



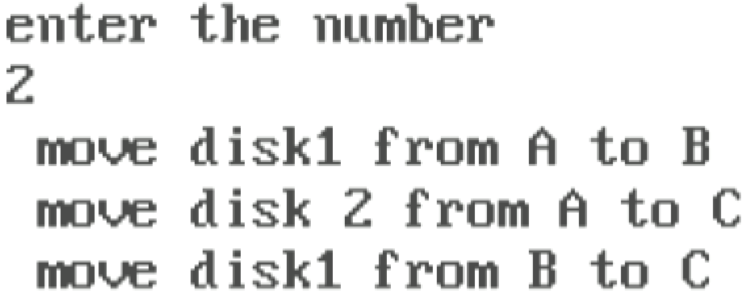
```
enter the range
4
enter the element
3 5 1 7
element in sorted are
1
3
5
7
```

PROGRAM-2

Q: WRITE A C++ PROGRAM FOR TOWER OF HANNOI

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

void towers(int,char,char,char);

        void main()

{

        int n;

        clrscr();

        cout<<"enter the number"<<endl;

        cin>>n;

        towers(n,'A','B','C');

        getch();

}

        void towers(int n,char source,char aux,char des)

{

        if(n==1)

   {

        cout<<" move disk1 from "<<source<<" to "<<des<<endl;

        return;

   }

        else

   {

        towers(n-1,source,des,aux);

        cout<<" move disk "<<n<<" from "<<source<<" to "<<des<<endl;

        towers(n-1,aux,source,des);

   }}
```

OUTPUT



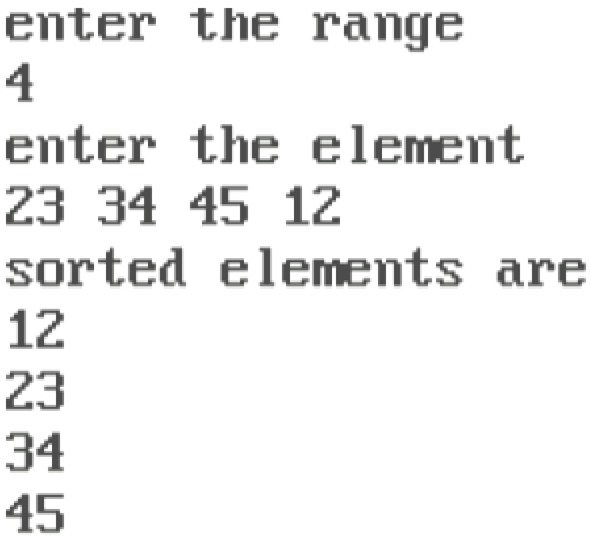
```
enter the number
2
 move disk1 from A to B
 move disk 2 from A to C
 move disk1 from B to C
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-3

Q: WRITE A C++ PROGRAM TO DEMONSTRATE INSERTION SORT

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

        int i,temp,j,n,a[15];

  class insert

{

   public:

          void getdata();

          void sort();

          void display();

};

        void insert::getdata()

{

        cout<<"enter the range"<<endl;

        cin>>n;

        cout<<"enter the element"<<endl;

        for(i=0;i<n;i++)

        {

        cin>>a[i];

        }

}

        void insert::sort()

{

        for(i=1;i<=n-1;i++)

{

        j=i;

        temp=a[j];
        while(j>0&&(a[j-1]>temp))
{
        a[j]=a[j-1];
        j--;
}
        a[j]=temp;
}
}
        void insert::display()
{
        cout<<"sorted elements are"<<endl;
        for(i=0;i<n;i++)
        {
        cout<<a[i]<<endl;
        }
}
   void main()
{
        insert t;
        clrscr();
        t.getdata();
        t.sort();
        t.display();
        getch();
}
```

OUTPUT



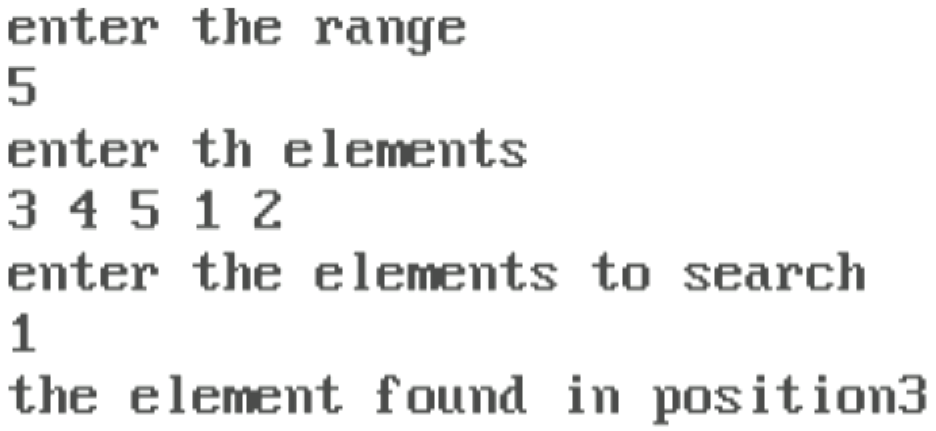
```
enter the range
4
enter the element
23 34 45 12
sorted elements are
12
23
34
45
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-04

Q: WRITE A C++ PROGRAM TO SEARCH AN ELEMENT USING LINEAR SEARCH TECHNIQUE

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

  void main()

{

     int a[20],i,j,n,pos,key,flag=0;

     clrscr();

     cout<<"enter the range"<<endl;

     cin>>n;

     cout<<"enter th elements"<<endl;

     for(i=0;i<n;i++)

      cin>>a[i];

      cout>>"enter the elements to search"<<endl;

      cin>>key;

     for(i=0;i<n;i++)

     if(key==a[i])

  {

     flag=1;

     pos=i;

     break;

  }

     if(flag==1)

      cout<<"the element found in position"<<pos<<endl;

     else

      cout<<"the element not found in the list"<<endl;

     getch();

}
```

OUTPUT:



```
enter the range
5
enter th elements
3 4 5 1 2
enter the elements to search
1
the element found in position3
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-5

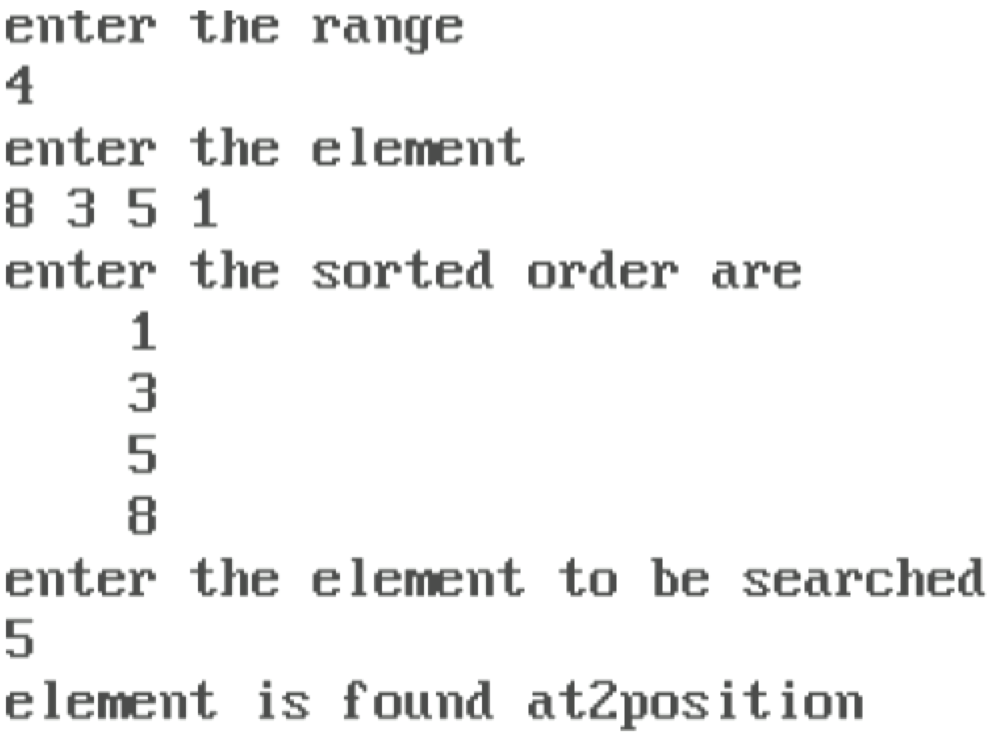
Q: WRITE A C++ PROGRAM TO DEMONSTRATE BINARY SEARCH

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

  void getdata();

  int a[15],n;

  void sort();

  void bsearch();

  void getdata()

{

        int i;

        cout<<"enter the range"<<endl;

        cin>>n;

        cout<<"enter the element"<<endl;

        for(i=0;i<n;i++)

        cin>>a[i];

}

   void sort()

{

        int i,j,temp;

        for(i=1;i<=n-1;i++)

   {

        for(j=0;j<=n-i-1;j++)

        if(a[j]>a[j+1])

         {

          temp=a[j];

          a[j]=a[j+1];

          a[j+1]=temp;
```

```
              }

     }

            cout<<"enter the sorted order are"<<endl;

            for(i=0;i<n;i++)

            cout<<setw(5)<<a[i]<<endl;

}

            void bsearch()

{

            int key,mid,flag=0,lb=0,ub=n-1;

            cout<<"enter the element to be searched"<<endl;

            cin>>key;

            lb=0;

            ub=n-1;

            while(lb<=ub)

    {

      mid=(lb+ub)/2;

      if(a[mid]==key)

    {

      flag=1;

      break;

    }

      if(key>a[mid])

      lb=mid+1;

    else

            ub=mid-1;

    }

      if(flag==1)

    {

      cout<<"element is found at"<<"position"<<endl;
```

```
 }
  else
        cout<<"element not found"<<endl;
 }
       void main()
 {
       clrscr();
       getdata();
       sort();
       bsearch();
       getch();
 }
```

OUTPUT:



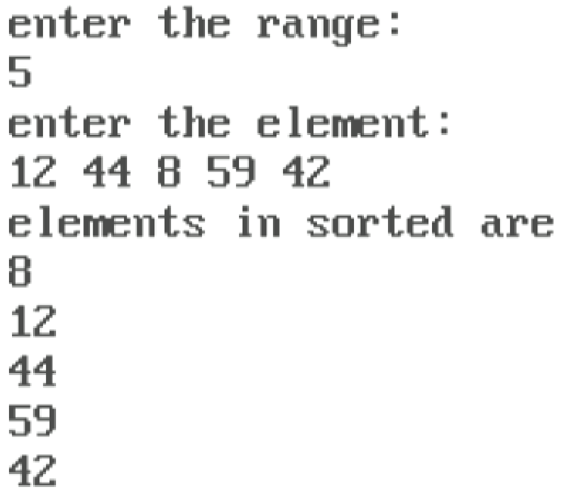
```
enter the range
4
enter the element
8 3 5 1
enter the sorted order are
    1
    3
    5
    8
enter the element to be searched
5
element is found at2position
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-6

Q : WRITE A C++ PROGRAM TO SORT THE GIVEN LIST USING SELECTION SORT TECHNIQUE

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

int i,j,temp,n,a[15];

class select

{

    public:

                void getdata();

                void sort();

                void display();

};

            void select::getdata()

{

            cout<<"enter the range:"<<endl;

            cin>>n;

            cout<<"enter the element:"<<endl;

            for(i=0;i<n;i++)

            cin>>a[i];

}

            void select::sort()

{

            int i,j,pos,temp;

            for(i=0;i<=n;i++)

    {

            pos=i;

            for(j=i+1;j<n-1;j++)

            if(a[j]<=a[pos])
    {

         pos=j;

    }

         temp=a[i];

         a[i]=a[pos];

         a[pos]=temp;

}

}

        void select::display()

{

        cout<<"elements in sorted are"<<endl;

        for(i=0;i<n;i++)

        cout<<a[i]<<endl;

}

   void main()

{

   select s;

   clrscr();

   s.getdata();

   s.sort();

   s.display();

   getch();

}
```

OUTPUT:



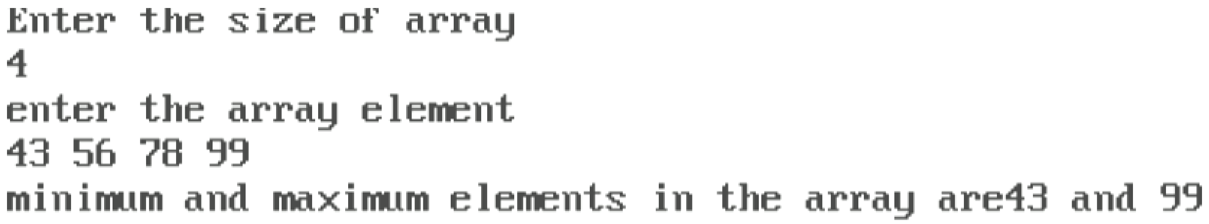
```
enter the range:
5
enter the element:
12 44 8 59 42
elements in sorted are
8
12
44
59
42
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-7

Q : WRITE A C++ PROGRAM TO IMPLEMENT DYNAMIC ARRAY ALSO FIND THE SMALLEST AND LARGEST ARRAYS.

```
#include<conio.h>
#include<iostream.h>
#include<stdlib.h>
void main()
{
int n,i,*arr,max,min;
clrscr();
cout<<"Enter the size of array"<<endl;
cin>>n;
arr=(int*)calloc(n,sizeof(int));
cout<<"enter the array element"<<endl;
for (i=0;i<n;i++)
{
cin>>arr[i];
}
max=arr[0];
for(i=0;i<n;i++)
{
if(max>arr[i])
{
max=arr[i];
}
}
min=arr[0];
for(i=0;i<n;i++)
{
if(min<arr[i])
{
min=arr[i];
}
}
cout<<"maximum and minimum elements in the array are"<<max<<"and"<<min;
getch();
}
```

OUTPUT:



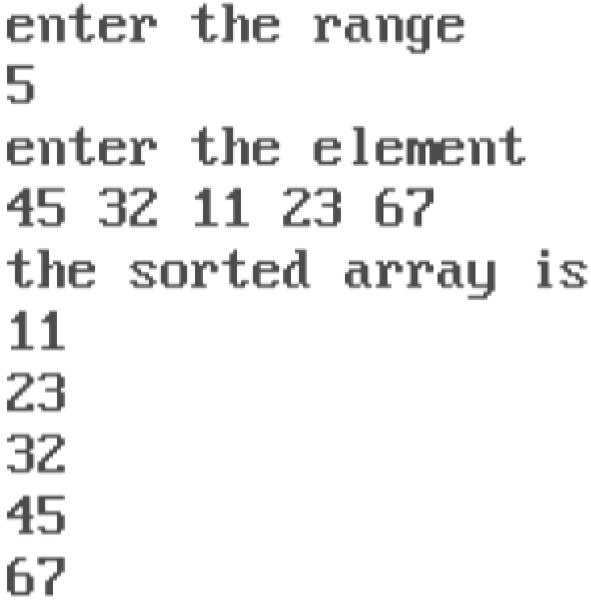
```
Enter the size of array
4
enter the array element
43 56 78 99
minimum and maximum elements in the array are43 and 99
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-8

Q : WRITE A C++ PROGRAM TO DEMONSTRATE MERGE SORT

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

void mergesort(int a[],int,int);

void mergearray(int a[],int,int,int);

int a[20],n;

void main()

{

        int i,low,high,mid;

        clrscr();

        cout<<"enter the range"<<endl;

        cin>>n;

        cout<<"enter the element"<<endl;

        for(i=0;i<n;i++)

        cin>>a[i];

        low=0;

        high=n-1;

        mergesort(a,low,high);

        cout<<"the sorted array is"<<endl;

        for(i=0;i<n;i++)

        cout<<a[i]<<endl;

        getch();

}

        void mergesort(int a[],int low,int high)

{

        int mid;

        if(low<high)
{
        mid=(low+high)/2;
        mergesort(a,low,mid);
        mergesort(a,mid+1,high);
        mergearray(a,low,mid,high);
}
}
        void mergearray(int a[],int low,int mid,int high)
{
        int c[20],i,j,k;
        i=low;
        j=mid+1;
        k=low;
        while((i<=mid)&&(j<=high))
        if(a[i]<a[j])
        c[k++]=a[i++];
        else c[k++]=a[j++];
        while(i<=mid)
        c[k++]=a[i++];
        while(j<high)
        c[k++]=a[j++];
        for(i=1000;i<=high;i++)
        a[i]=c[i];
}
```

OUTPUT:



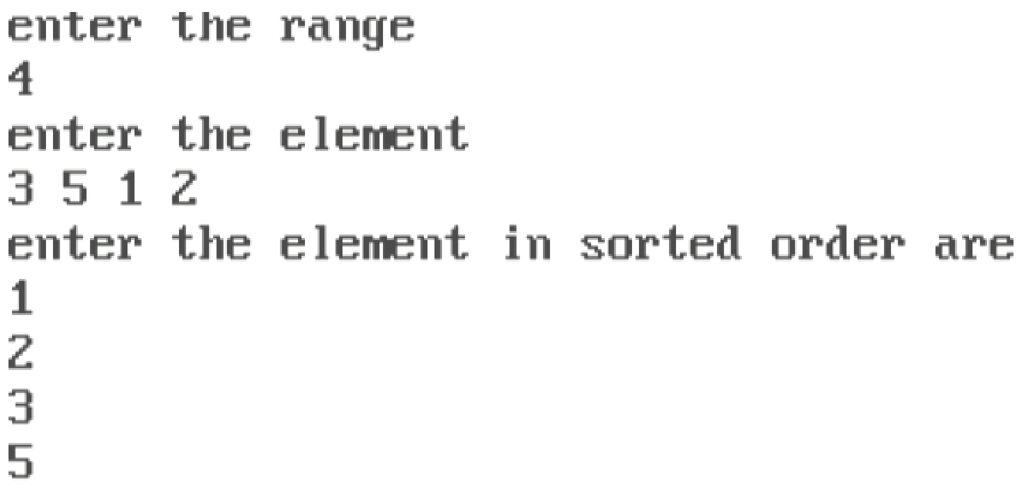
```
enter the range
5
enter the element
45 32 11 23 67
the sorted array is
11
23
32
45
67
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-9

Q : WRITE A C++ PROGRAM TO DEMONSTRATE QUICK SORT

```
#include<iostream.h>

#include<conio.h>

#include<iomanip.h>

  int a[50],n;

  int partition(int a[],int,int);

  void quicksort(int a[],int,int);

   void main()

{

        int i;

        clrscr();

        cout<<"enter the range"<<endl;

        cin>>n;

        cout<<"enter the element"<<endl;

        for(i=0;i<n;i++)

        cin>>a[i];

        quicksort(a,0,n-1);

        cout<<"enter the element in sorted order are"<<endl;

        for(i=0;i<n;i++)

        cout<<a[i]<<endl;

        getch();

}

    void quicksort(int a[],int low,int high)

{

        int loc;

        if(low<high)

    {

        loc=partition(a,low,high);
        quicksort(a,low,loc-1);

        quicksort(a,loc+1,high);

    }

}

        int partition(int a[],int low,int high)

{

        int pivot,i,j,temp;

        pivot=a[low];

        i=low+1;

        j=high;

        if(i<j)

    {

        while(i<j)

    {

        while(a[i]<=pivot)

    {

        i++;

    }

        while (a[i]>pivot)

    {

        j--;

    }

        if(i<j)

    {

        temp=a[i];

        a[i]=a[j];

        a[j]=temp;

    }

        else
    {

        temp=a[low];

        a[low]=a[j];

        a[j]=temp;

    }

    }

    }

        return(j);

}
```

OUTPUT:



```
enter the range
4
enter the element
3 5 1 2
enter the element in sorted order are
1
2
3
5
```

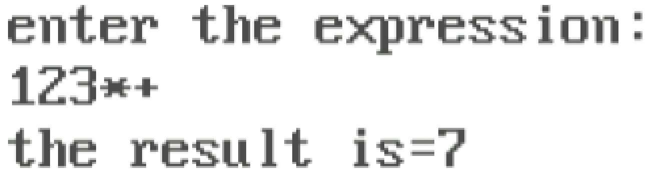
ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-10

Q : WRITE A C++ PROGRAM TO EVALUATE POSTFIX EXPRESSION

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<ctype.h>
int s[10],top=-1;
void push(int);
int pop();
void push(int x)
{
 s[++top]=x;
 }
int pop()
{
  return(s[top--]);
}
void main()
{
  char expr[20],ch;
  int op1,op2,n;
  clrscr();
  cout<<"enter the expression:"<<endl;
  cin>>expr;
  n=strlen(expr);
 for(int i=0;i<n;i++)
  {
     ch=expr[i];
     if(isdigit(ch))
     push(ch-'0');
     else
     {
       op2=pop();
       op1=pop();
       switch(ch)
       {
       case '+':{
                              push(op2+op1);
                              break;
                            }
       case '-':{
                            push(op2-op1);
                            break;
                            }
       case '*':{
                             push(op2*op1);
                             break;
```

```
                  }
   case '/':{
                 push(op2/op1);
                 break;
                 }
   }
  }
 }
cout<<"the result is="<<s[top]<<endl;
getch();
}
```

OUTPUT:



```
enter the expression:
123*+
the result is=7
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-11

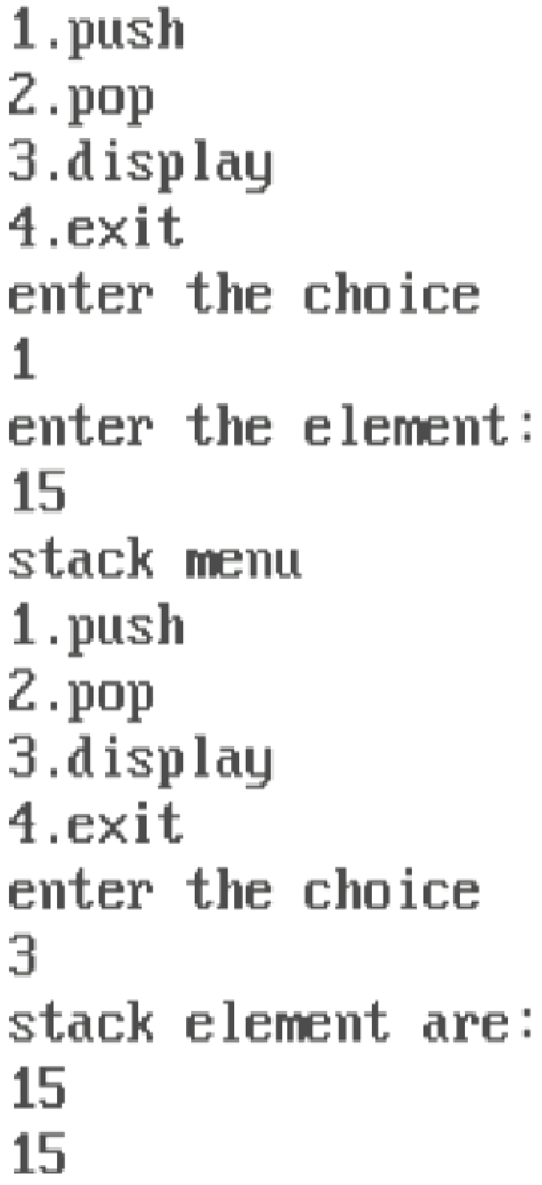
Q : WRITE A C++ PROGRAM TO IMPLEMENT STACK OPERATIONS USING ARRAYS

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<stdlib.h>
#define size 5
  class stack
{
        private:int s[size],n,i,top;
        public:stack();
                void push();
                void pop();
                void display();
};
        stack::stack()
{
        top=-1;
}
        void stack::push()
{
        if(top>=size-1)
        {
         cout<<"""stack is overflow"<<endl;
         return;
        }
        top++;
        cout<<"""enter the element:"<<endl;
        cin>>n;
```

```
          s[top]=n;

}

        void stack::pop()

{

        if(top==-1)

    {

        cout<<"""stack is underflow"<<endl;

        return;

    }

        n=s[top];

        top--;

        cout<<"""the deleted element is:"<<n<<endl;

}

        void stack::display()

{

        if(top==-1)

    {

        cout<<"""stack is empty"<<endl;

        return;

    }

        cout<<"""stack element are:"<<endl;

        for(i=top;i>=0;i--)

        cout<<s[i]<<endl;

}

   void main()

{

        stack s;

        clrscr();

        int ch;
```

```
            do
    {
      cout<<"""stack menu"<<endl;
      cout<<"""1.push"<<endl;
      cout<<"""2.pop"<<endl;
      cout<<"""3.display"<<endl;
      cout<<"""4.exit"<<endl;
      cout<<"""enter the choice"<<endl;
      cin>>ch;
   switch(ch)
     {
    case1:{
                s.push();
                break;
              }
    case2:{
                s.pop();
                break;
              }
    case3:{
                s.display();
                break;
              }
    case4:{
                exit(0);
                getch();
              }
    default:{
                 cout<<"""invalid choice"<<endl;
                }
               }
              }
               while(ch<=4);


}
```

OUTPUT:



```
1.push
2.pop
3.display
4.exit
enter the choice
1
enter the element:
15
stack menu
1.push
2.pop
3.display
4.exit
enter the choice
3
stack element are:
15
15
```

ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-12

Q : WRITE A C++ PROGRAM TO IMPLEMENT QUEUE OPERATIONS

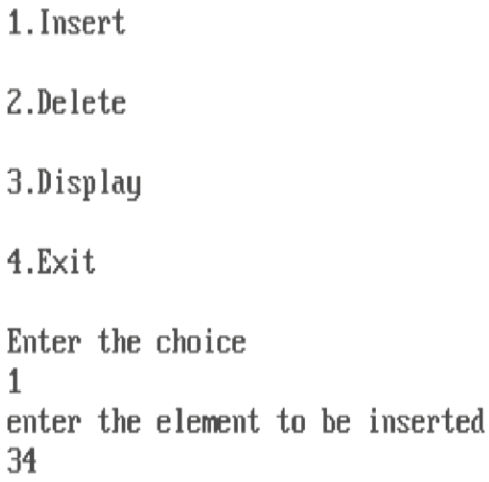
```
#include<iostream.h>
 #include<iomanip.h>
 #include<conio.h>
 #include<stdlib.h>
 # define size 5

 class queue
 {
 private:int q[size],front,rear;
 public:queue();
 void qinsert();
 void qdelete();
 void qdisplay();
 };

queue::queue()
{
front=-1;
rear=-1;
}
void queue::qinsert()
{
int num;
if(rear==size-1)
{
cout<<"queue overflow"<<endl;
return;
```

```
}
rear++;
cout<<"enter the element to be inserted"<<endl;
cin>>num;
q[rear]=num;
if(front==-1)
{
front=0;
return;
}
}
void queue::qdelete()
{
int num;
if(front==-1)
{
cout<<"queue underflow"<<endl;
return;
}
num=q[front];
cout<<"the deleted element is" <<endl<<num<<endl;
if(front==rear)
{
front=-1;
rear=-1;
}
else
{
front++;
```

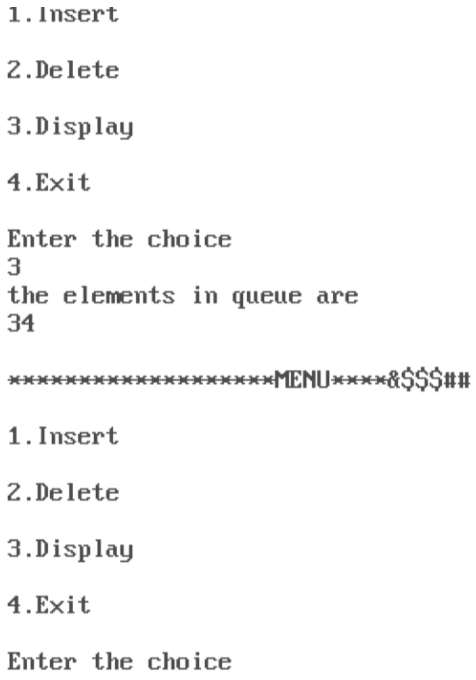
```
return;

}

}

void queue::qdisplay()

{

int i;

if(front==-1 && rear==-1)

{

cout<<"queue is empty"<<endl;

return;

}

cout<<"the elements in queue are"<<endl;

for(i=front;i<=rear;i++)

cout<<q[i]<<"\t"<<endl;

}

void main()

{

queue q;

int ch;

clrscr();

do

{

cout<<endl<<"*******************MENU****&$$$##"<<endl;

cout<<endl<<"1.Insert"<<endl;

cout<<endl<<"2.Delete"<<endl;

cout<<endl<<"3.Display"<<endl;

cout<<endl<<"4.Exit"<<endl;

cout<<endl<<"Enter the choice"<<endl;

cin>>ch;
```

```
switch(ch)
{
case 1:{
            q.qinsert();
            break;
            }
case 2:{
    q.qdelete();
    break;
    }
    case 3:{
    q.qdisplay();
    break;
    }
    case 4:
    {
    exit(0);
    }
    default:{
    cout<<"invalid choice"<<endl;

    }
    }
    }
    while(ch<=4);
    getch();
    }
```

OUTPUT:



```
1.Insert

2.Delete

3.Display

4.Exit

Enter the choice
1
enter the element to be inserted
34
```



```
1.Insert

2.Delete

3.Display

4.Exit

Enter the choice
3
the elements in queue are
34

*******************MENU****&$$$##

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice
```

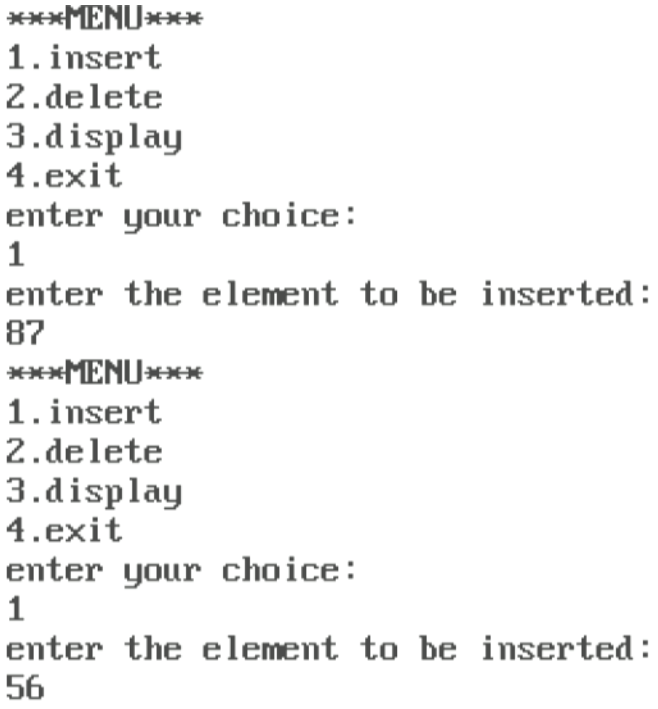
ABHINAV NARAYANAN
3SU21CC007
LAB ON DS
PROGRAM-13

Q : WRITE A C++ PROGRAM TO IMPLEMENT A CIRCULAR QUEUE USING AN ARRAY.

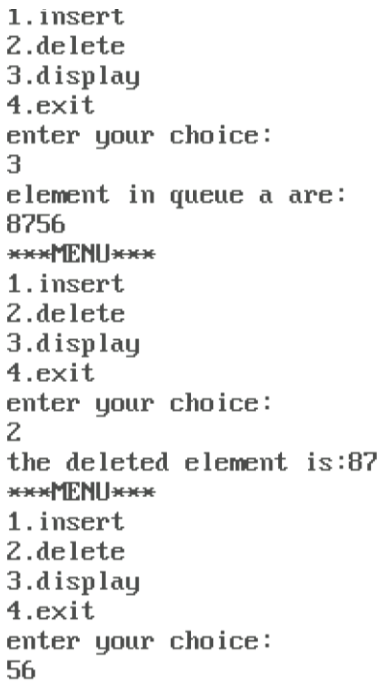
```
#include<iostream.h>

#include<iomanip.h>

#include<conio.h>

#include<stdlib.h>

#define max 5

class cqueue

{

  private:int cq[max],front,rear;

  public:void cqinsert();

  void cqdelete();

  void cqdisplay();

   cqueue()

  {

   front=rear=-1;

  }

};

void cqueue::cqinsert()

{

 int num;

 if((rear+1)%max==front||(front==0&&rear==max-1))

 {

  cout<<"circular queue overflow"<<endl;

  return;

 }

 else

 {

  cout<<"enter the element to be inserted:"<<endl;
```

```
    cin>>num;

    if(front==-1)

     front=0;

     rear=(rear-1)%max;

     cq[rear]=num;

    }

  }

void cqueue::cqdelete()

{

  int num;

  if(front==-1)

  {

    cout<<"circular queue underflow"<<endl;

    return;

  }

 num=cq[front];

 if(front==rear)

 {

  front=-1;

  rear=-1;

 }

 else

 {

  front=(front+1)%max;

 }

 cout<<"the deleted element is:"<<num<<endl;

 }

 void cqueue::cqdisplay()

 {
  int i=front;
  if(front==-1)
  {
    cout<<"circular queue is empty"<<endl;
    }
    else
    {
    cout<<"element in queue a are:"<<endl;
    while(i<=rear)
    {
     cout<<cq[i]<<"";
     i=(i+1)%max;
     }
     cout<<endl;
    }
  }
void main()
{
 cqueue c;
 clrscr();
 int ch;
 do
 {
   cout<<"***MENU***"<<endl;
   cout<<"1.insert"<<endl;
   cout<<"2.delete"<<endl;
   cout<<"3.display"<<endl;
   cout<<"4.exit"<<endl;
   cout<<"enter your choice:"<<endl;
   cin>>ch;

   switch(ch)

   {

   case 1:{

   c.cqinsert();

   break;

   }

   case 2:{

   c.cqdelete();

   break;

   }

   case 3:{

   c.cqdisplay();

   break;

   }

   case 4:{

   exit(0);

   }

   default:{

   cout<<"invalid choice"<<endl;

   }

 }

}

while(ch<=4);

getch();

}
```

OUTPUT:



```
***MENU***
1.insert
2.delete
3.display
4.exit
enter your choice:
1
enter the element to be inserted:
87
***MENU***
1.insert
2.delete
3.display
4.exit
enter your choice:
1
enter the element to be inserted:
56
```



```
1.insert
2.delete
3.display
4.exit
enter your choice:
3
element in queue a are:
8756
***MENU***
1.insert
2.delete
3.display
4.exit
enter your choice:
2
the deleted element is:87
***MENU***
1.insert
2.delete
3.display
4.exit
enter your choice:
56
```