DS Practical's

1. Assignment Name: Demonstration of Array

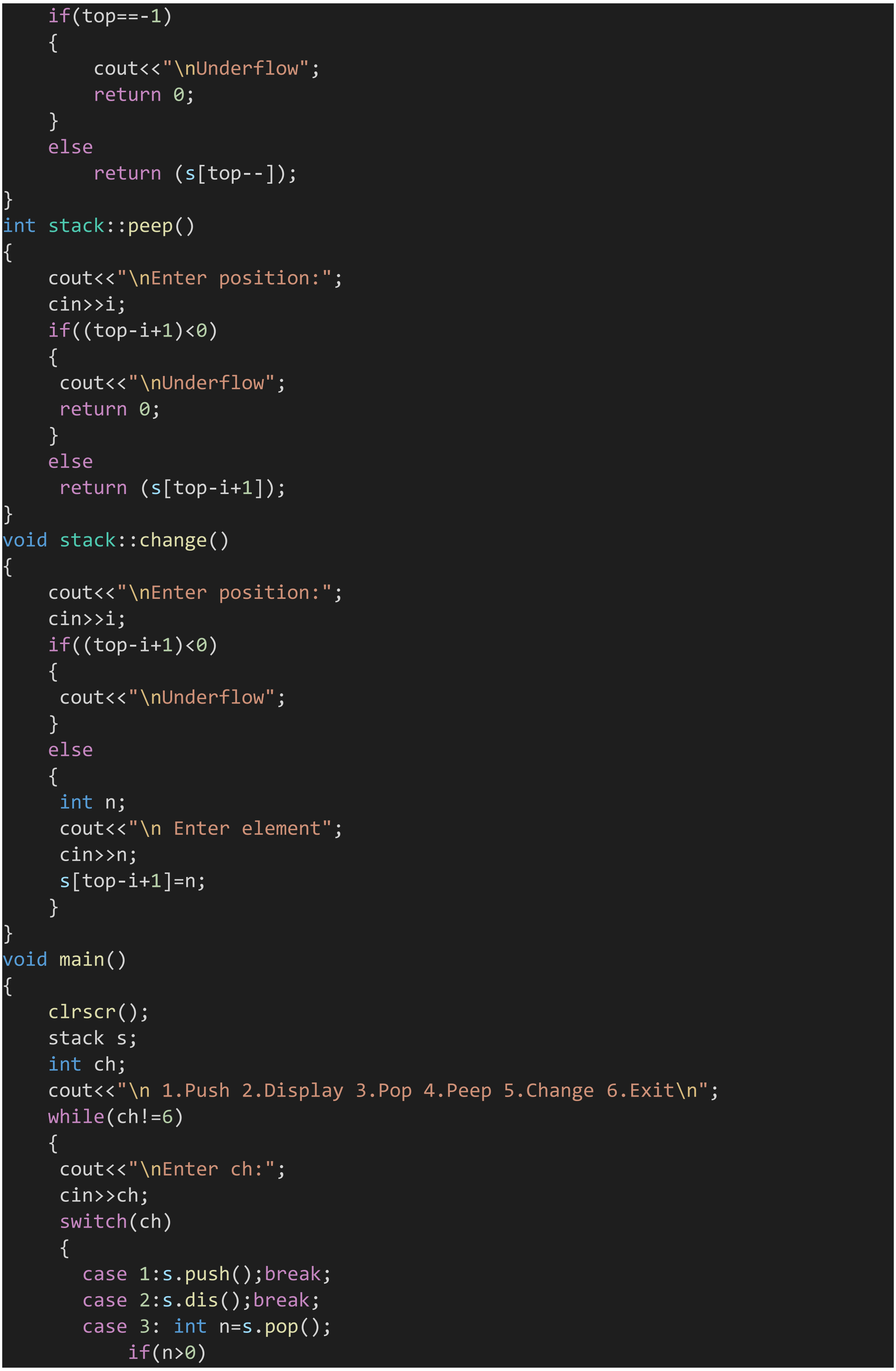
```
#include<iostream.h>

#include<conio.h>
#include<process.h>
class demo
{
    int a[10],i,j,n,item,k;
    public:
          void get();
          void insert();
          void del();
          void dis();

};
 void demo::get()
 {
 cout<<"\Enter n";
 cin>>n;
 cout<<"\n Enter Array Elements:";
 for(i=1;i<=n;i++)
 cin>>a[i];
}
void demo::insert()
{
cout<<"\n Enter Position :";
cin>>k;
cout<<"\n Enter Item:";
cin>>item;
j=n;
while(j>=k)
{
a[j+1]=a[j];
j--;
}
a[k]=item;
n++;
}
void demo::del()
{
cout<<"\n Enter Position;";
cin>>k;
j=k;
while(j<=n-1)
{
a[j]=a[j+1];
j++;
}
n--;
}
void demo::dis()
{
cout<<"\n Elements are \n";
for(i=1;i<=n;i++)
cout<<a[i]<<"\t";
}



void main()
{
clrscr();
demo d;
int ch;
d.get();
cout<<"\n1. Insert 2.Del 3.Dis 4. Exit \n";
while(ch!=4)
{
cout<<"\n Enter choice";
cin>>ch;
switch(ch)
{
    case 1:d.insert();break;
    case 2:d.del();break;
    case 3:d.dis();break;
    case 4:exit(0);
    }
}
getch();
}
```

2. Assignment Name: Demonstration of Matrix

```
#include<iostream.h>

#include<conio.h>

class matrix
{
    int a[5][5],b[5][5],c[5][5],d[5][5],e[5][5],f[5][5];
    int p,q,i,j,k,n,m;

    public:
        void get();
        void add();
        void sub();
};
void matrix::get()
{
    cout<<"\n Enter number of row and column:\t";
    cin>>n>>m;

    cout<<"\n Enter the first matrix: \n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        cin>>a[i][j] ;
    }

        cout<<"\n Enter number of row and column:\t";
    cin>>p>>q;

    cout<<"\n Enter the second matrix: \n";
    for(i=0;i<p;i++)
    {
        for(j=0;j<q;j++)
        cin>>b[i][j] ;
    }
}
void matrix::add()
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
    }
    cout<<"\n The addition of two matrix is : \n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cout<<c[i][j]<<"\t";
            cout<<"\n";
    }
}
void matrix::sub()
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            d[i][j]=a[i][j]-b[i][j];
        }
    }
    cout<<"\n The substraction of two matrix is : \n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cout<<d[i][j]<<"\t";
            cout<<"\n";
    }
}
void main()
{
    clrscr();
    matrix m;
    m.get();
    m.add();
    m.sub();
    getch();
}
```

3. Assignment Name: Implementation of Stack for Integer

```
#include<iostream.h>

#include<conio.h>
#include<process.h>
class stack
{
    int s[10],n ,top,ele,i;
public:
    stack()
    {
       top=-1;
    }
    void push();
    void dis();
    int pop();
    int peep();
    void change();
};
void stack::push()
{
    if(top>=2)
    cout<<"\n Stack is overflow:";
    else
    {
    cout<<"\n Enter element:";
    cin>>ele;
    top++;
    s[top]=ele;
    }
}
void stack::dis()
{
    cout<<"\nElement in stack are:\n";
    for(i=top;i>=0;i--)
    cout<<s[i]<<"\t";
}
int stack::pop()
{
```



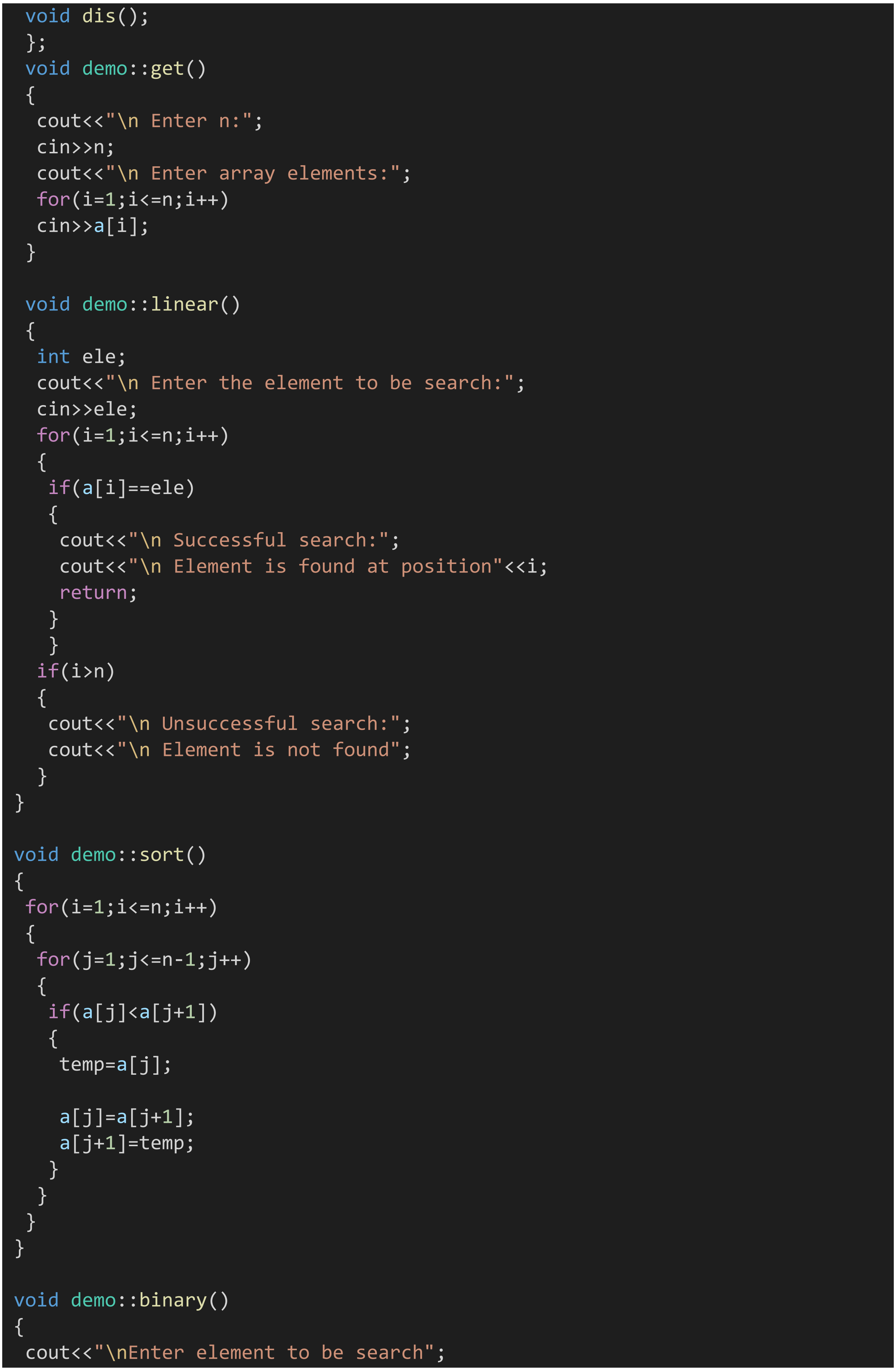
```
    if(top==-1)
    {
        cout<<"\nUnderflow";
        return 0;
    }
    else
        return (s[top--]);
}
int stack::peep()
{
    cout<<"\nEnter position:";
    cin>>i;
    if((top-i+1)<0)
    {
     cout<<"\nUnderflow";
     return 0;
    }
    else
     return (s[top-i+1]);
}
void stack::change()
{
    cout<<"\nEnter position:";
    cin>>i;
    if((top-i+1)<0)
    {
     cout<<"\nUnderflow";
    }
    else
    {
     int n;
     cout<<"\n Enter element";
     cin>>n;
     s[top-i+1]=n;
    }
}
void main()
{
    clrscr();
    stack s;
    int ch;
    cout<<"\n 1.Push 2.Display 3.Pop 4.Peep 5.Change 6.Exit\n";
    while(ch!=6)
    {
     cout<<"\nEnter ch:";
     cin>>ch;
     switch(ch)
     {
       case 1:s.push();break;
       case 2:s.dis();break;
       case 3: int n=s.pop();
           if(n>0)
           cout<<"\nPop ele is "<<n;
           break;
       case 4: int m=s.peep();
           if(m>0)
           cout<<"\nPeep ele is "<<m;
           break;
       case 5: s.change();break;
       case 6: exit(0);
     }
       }
       getch();
}
```

4. Assignment Name: Implementation of Infix to Postfix Expression

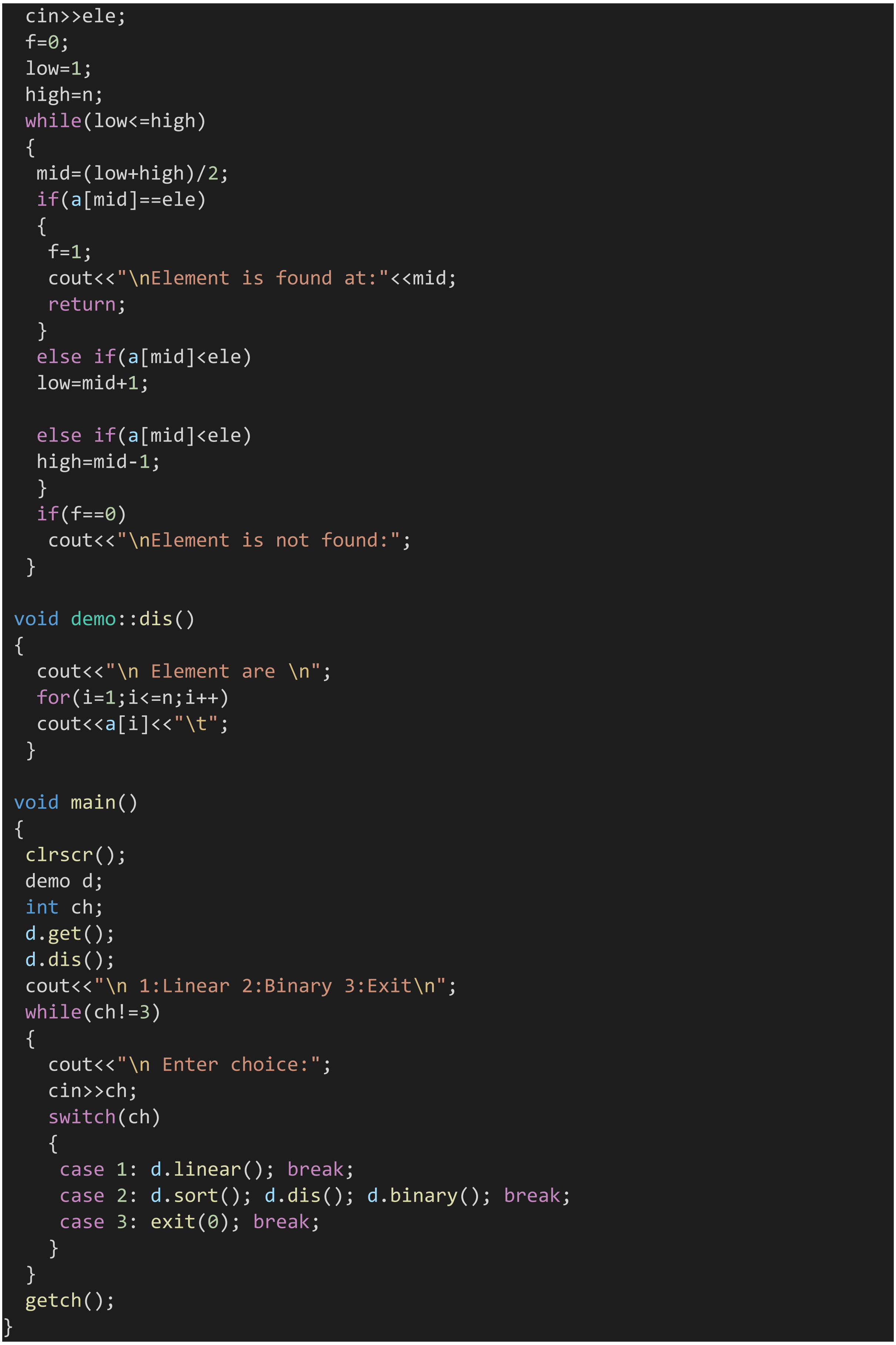
```
#include<iostream.h>

#include<conio.h>
#include<string.h>
class convert
{
     char infix[20],postfix[20],s[20];
     int i,p,top;
public:
     convert()
     {
    top=-1;
    i=p=0;
    cout<<"\nEnter infix Expression: ";
    cin>>infix;
    strcat(infix,")");
    s[++top]='(';
     }
     int precedance(char);
     void post();
     void display();
};
int convert::precedance(char ch)
{
    switch(ch)
    {
    case '^':return 3;
    case '*':return 2;
    case '/':return 2;
    case '+':return 1;
    case '-':return 1;
    default: return 0;
     }
}
void convert::post()
{
    char ch;
    while(top!=-1)
    {
       ch=infix[i++];
       if((ch>='A'&&ch<='Z')||(ch>='a'&&ch<='z')||(ch>='1'&&ch<='9'))
        postfix[p++]=ch;
       else if(ch=='(')
        s[++top]=ch;
       else if(ch=='+'||ch=='-'||ch=='*'||ch=='/'||ch=='^')
       {
        while(precedance(ch)<=precedance(s[top]))
        postfix[p++]=s[top--];
        s[++top]=ch;
       }
       else if(ch==')')
       {
        while(s[top]!='(')
        postfix[p++]=s[top--];
        top--;
       }
       else
       cout<<"\nWrong string";
       }
    postfix[p]='\0';
}
void convert::display()
{
    cout<<"\nPostfix Expression is : "<<postfix;
}
void main()
{
     clrscr();
     convert c;
     c.post();
     c.display();
     getch();
}
```

5. Assignment Name: Implementation of Linear and Binary Search

```
#include<iostream.h>

#include<conio.h>
#include<process.h>

class demo
{
 int a[10],i,j,n,f,temp,ele,demo,mid,low,high;
 public:
  void get();
  void sort();
  void linear();
  void binary();
```



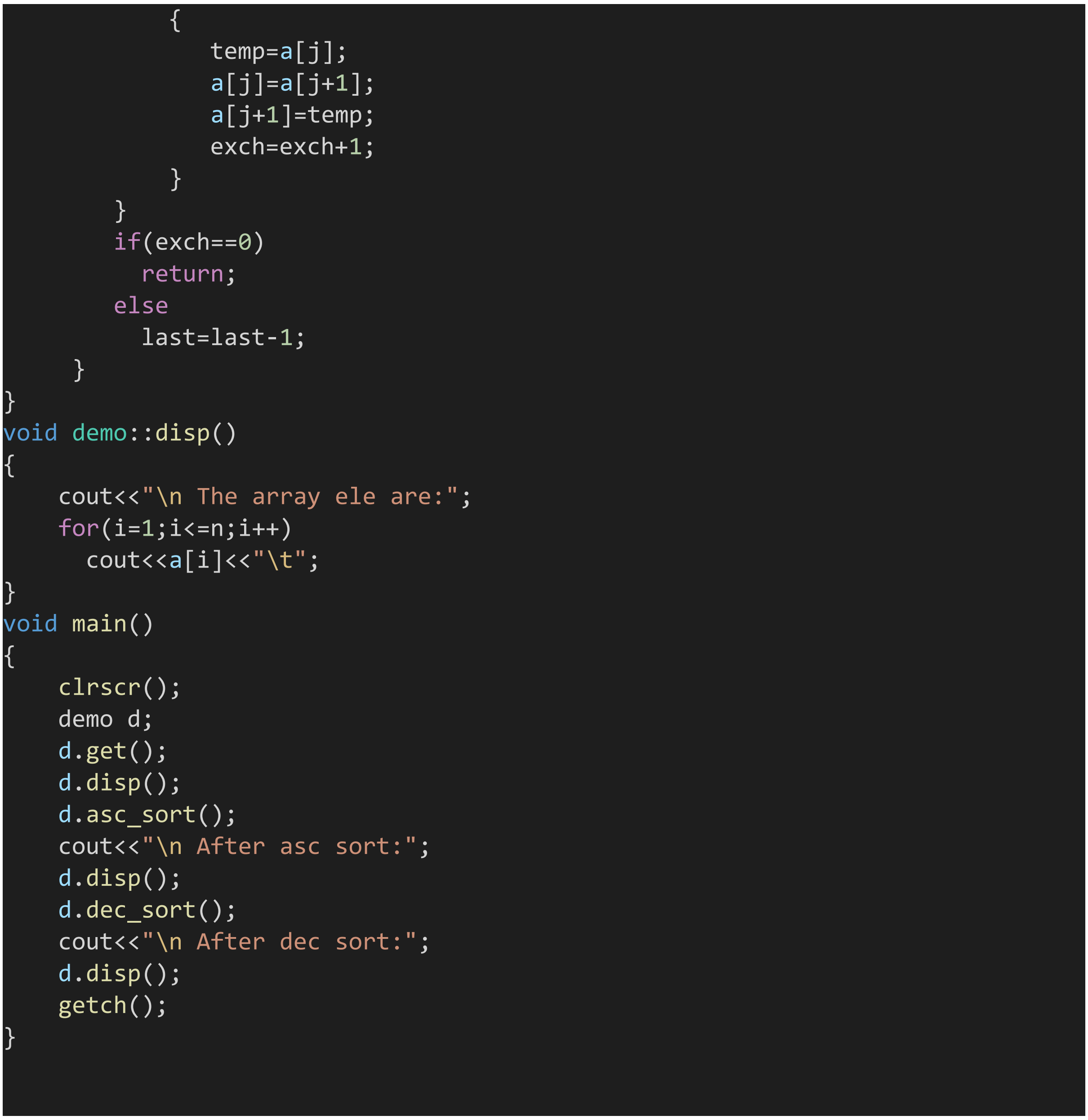
```
 void dis();
 };
 void demo::get()
 {
  cout<<"\n Enter n:";
  cin>>n;
  cout<<"\n Enter array elements:";
  for(i=1;i<=n;i++)
  cin>>a[i];
 }

 void demo::linear()
 {
  int ele;
  cout<<"\n Enter the element to be search:";
  cin>>ele;
  for(i=1;i<=n;i++)
  {
   if(a[i]==ele)
   {
    cout<<"\n Successful search:";
    cout<<"\n Element is found at position"<<i;
    return;
   }
   }
  if(i>n)
  {
   cout<<"\n Unsuccessful search:";
   cout<<"\n Element is not found";
  }
}

void demo::sort()
{
 for(i=1;i<=n;i++)
 {
  for(j=1;j<=n-1;j++)
  {
   if(a[j]<a[j+1])
   {
    temp=a[j];

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

void demo::binary()
{
 cout<<"\nEnter element to be search";
```



```
  cin>>ele;
  f=0;
  low=1;
  high=n;
  while(low<=high)
  {
   mid=(low+high)/2;
   if(a[mid]==ele)
   {
    f=1;
    cout<<"\nElement is found at:"<<mid;
    return;
   }
   else if(a[mid]<ele)
   low=mid+1;

   else if(a[mid]<ele)
   high=mid-1;
   }
   if(f==0)
    cout<<"\nElement is not found:";
  }

 void demo::dis()
 {
   cout<<"\n Element are \n";
   for(i=1;i<=n;i++)
   cout<<a[i]<<"\t";
  }

 void main()
 {
  clrscr();
  demo d;
  int ch;
  d.get();
  d.dis();
  cout<<"\n 1:Linear 2:Binary 3:Exit\n";
  while(ch!=3)
  {
    cout<<"\n Enter choice:";
    cin>>ch;
    switch(ch)
    {
     case 1: d.linear(); break;
     case 2: d.sort(); d.dis(); d.binary(); break;
     case 3: exit(0); break;
    }
  }
  getch();
}
```

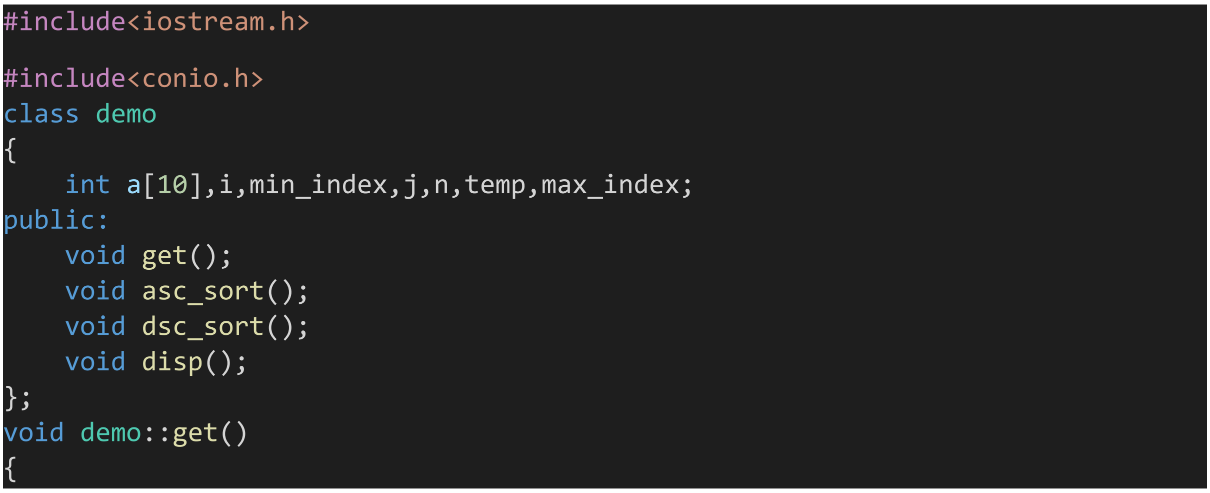
6. Assignment Name: Perform Bubble Sort for Integer

```
#include<iostream.h>

#include<conio.h>
class demo
{
    int a[10],i,last,exch,j,n,temp;
public:
    void get();
    void asc_sort();
    void dec_sort();
    void disp();
};
void demo::get()
{
    cout<<"\n Enter the array size:";
    cin>>n;
    cout<<"\n Enter the array ele:";
    for(i=1;i<=n;i++)
    cin>>a[i];
}
void demo::asc_sort()
{
    last=n;
    for(i=1;i<=n-1;i++)
    {
        exch=0;
        for(j=1;j<=last-1;j++)
        {
            if(a[j]>a[j+1])
            {
               temp=a[j];
               a[j]=a[j+1];
               a[j+1]=temp;
               exch=exch+1;
            }
        }
        if(exch==0)
          return;
        else
          last=last-1;
     }
}
void demo::dec_sort()
{
    last=n;
    for(i=1;i<=n-1;i++)
    {
        exch=0;
        for(j=1;j<=last-1;j++)
        {
            if(a[j]<a[j+1])
```

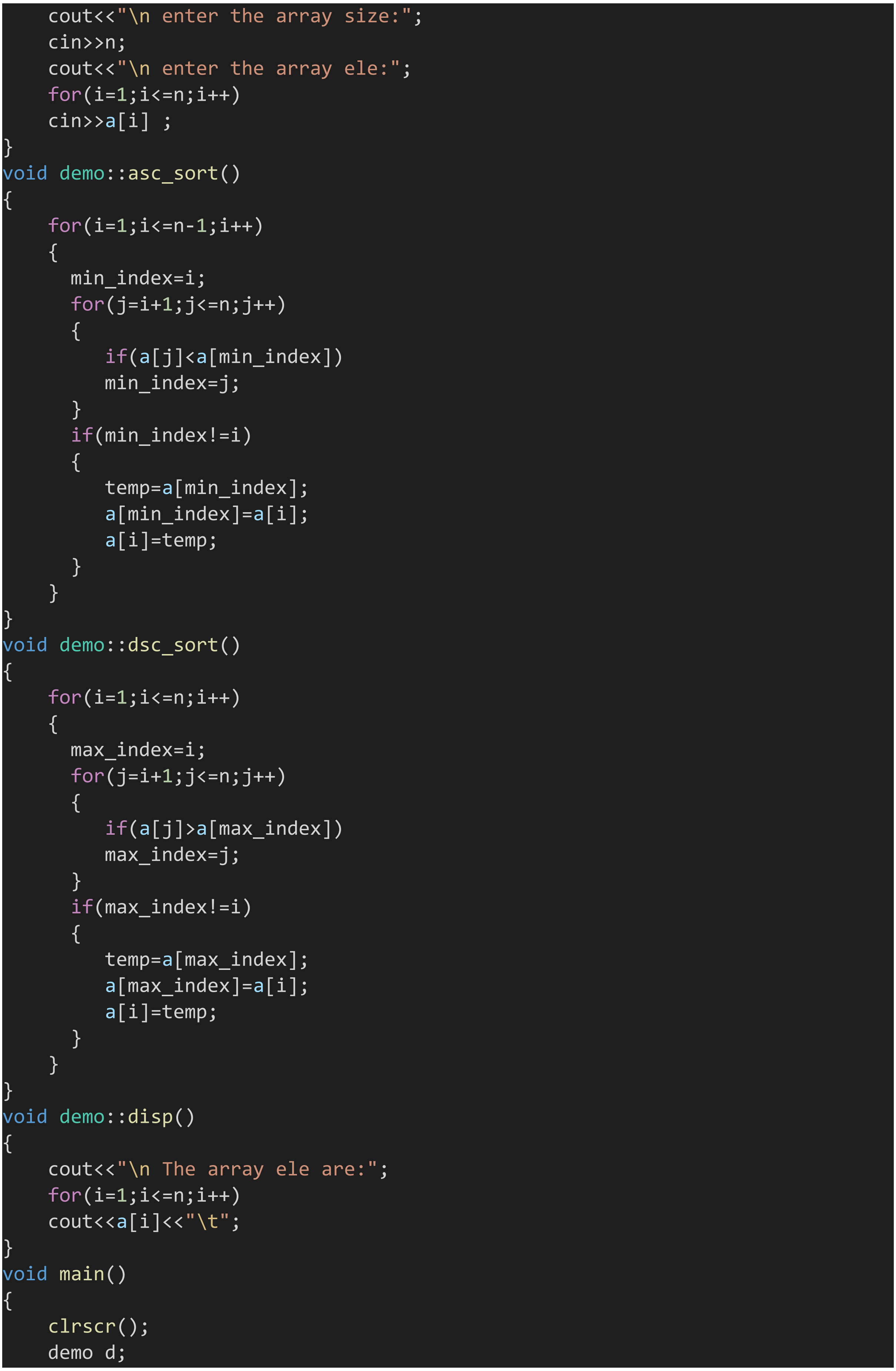


```
            {
               temp=a[j];
               a[j]=a[j+1];
               a[j+1]=temp;
               exch=exch+1;
            }
        }
        if(exch==0)
          return;
        else
          last=last-1;
     }
}
void demo::disp()
{
    cout<<"\n The array ele are:";
    for(i=1;i<=n;i++)
      cout<<a[i]<<"\t";
}
void main()
{
    clrscr();
    demo d;
    d.get();
    d.disp();
    d.asc_sort();
    cout<<"\n After asc sort:";
    d.disp();
    d.dec_sort();
    cout<<"\n After dec sort:";
    d.disp();
    getch();
}
```

7. Assignment Name: Implementation of Selection Sort



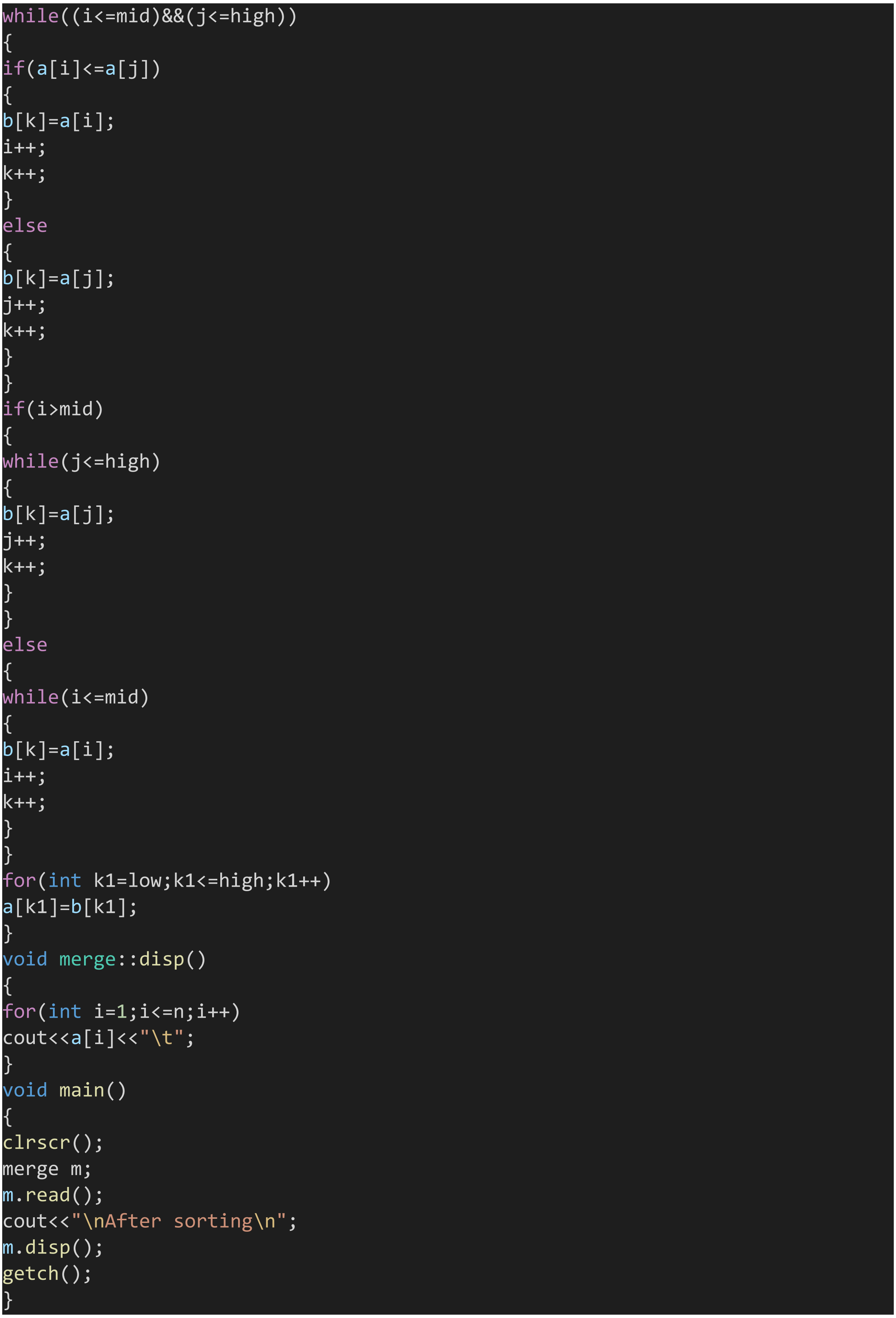
```
#include<iostream.h>

#include<conio.h>
class demo
{
    int a[10],i,min_index,j,n,temp,max_index;
public:
    void get();
    void asc_sort();
    void dsc_sort();
    void disp();
};
void demo::get()
{
```



```
    cout<<"\n enter the array size:";
    cin>>n;
    cout<<"\n enter the array ele:";
    for(i=1;i<=n;i++)
    cin>>a[i] ;
}
void demo::asc_sort()
{
    for(i=1;i<=n-1;i++)
    {
      min_index=i;
      for(j=i+1;j<=n;j++)
      {
         if(a[j]<a[min_index])
         min_index=j;
      }
      if(min_index!=i)
      {
         temp=a[min_index];
         a[min_index]=a[i];
         a[i]=temp;
      }
    }
}
void demo::dsc_sort()
{
    for(i=1;i<=n;i++)
    {
      max_index=i;
      for(j=i+1;j<=n;j++)
      {
         if(a[j]>a[max_index])
         max_index=j;
      }
      if(max_index!=i)
      {
         temp=a[max_index];
         a[max_index]=a[i];
         a[i]=temp;
      }
    }
}
void demo::disp()
{
    cout<<"\n The array ele are:";
    for(i=1;i<=n;i++)
    cout<<a[i]<<"\t";
}
void main()
{
    clrscr();
    demo d;
    d.get();
    d.disp();
    d.asc_sort();
    cout<<"\n After asc sort:";
    d.disp();
    d.dsc_sort();
    cout<<"\n After dsc sort:";
    d.disp();
    getch();
}
```

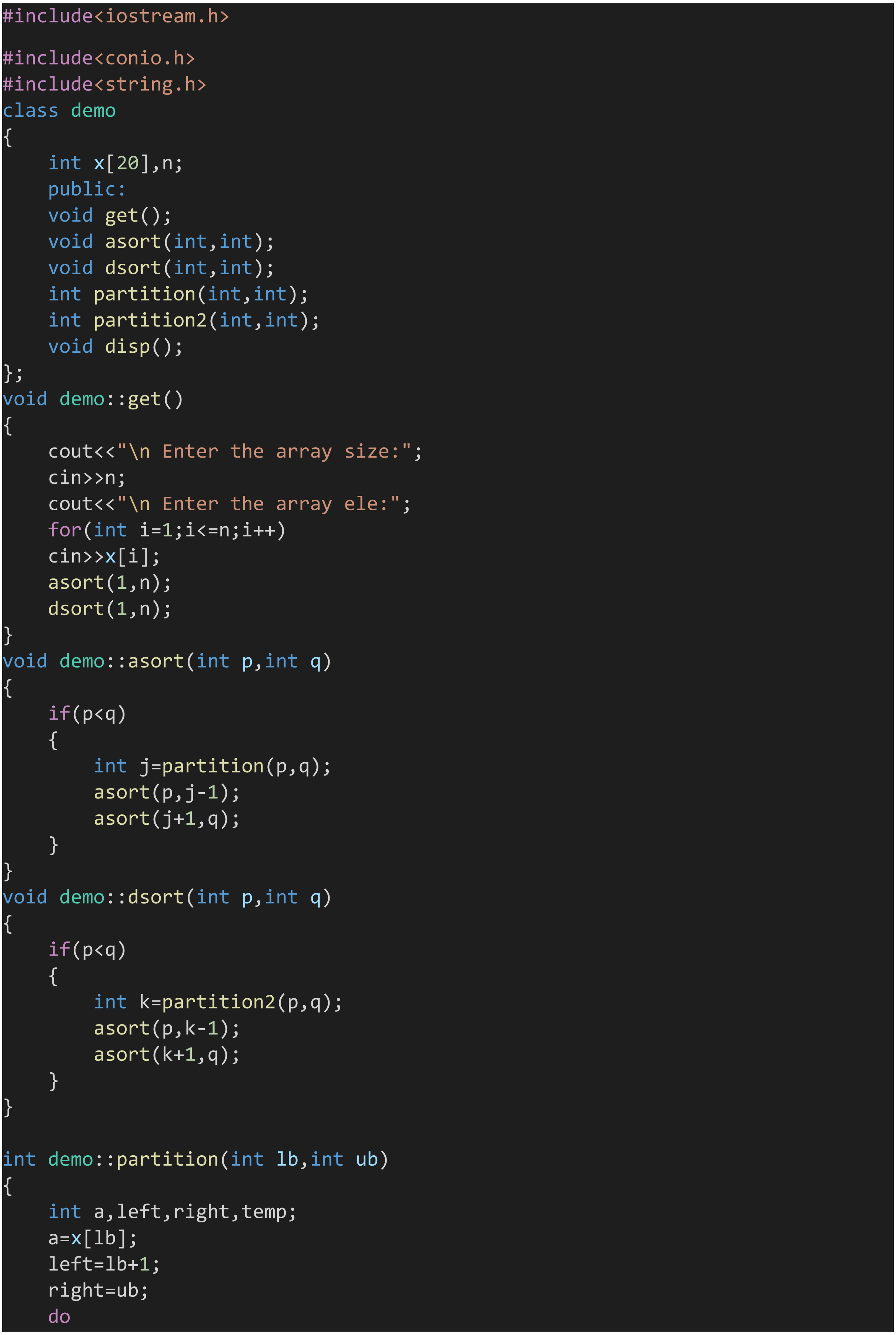
8. Assignment Name: Implementation of Merge sort

```
#include<iostream.h>

#include<conio.h>
#include<stdio.h>
class merge
{
int a[10],n;
public:
void read();
void merge_sort(int l,int h);
void mergel(int l,int m,int h);
void disp();
};
void merge::read()
{
cout<<"\n How many elements you want to store";
cin>>n;
cout<<"\n Enter elements \n";
for(int i=1;i<=n;i++)
cin>>a[i];
merge_sort(1,n);
}
void merge::merge_sort(int l, int h)
{
int mid;
if(l<h)
{
mid = int((l+h)/2);
merge_sort(l,mid);
merge_sort(mid+1,h);
mergel(l,mid,h);
}
}
void merge::mergel(int low,int mid,int high)
{
int b[10];
int i=low;
int k=low;
int j=mid+1;
```

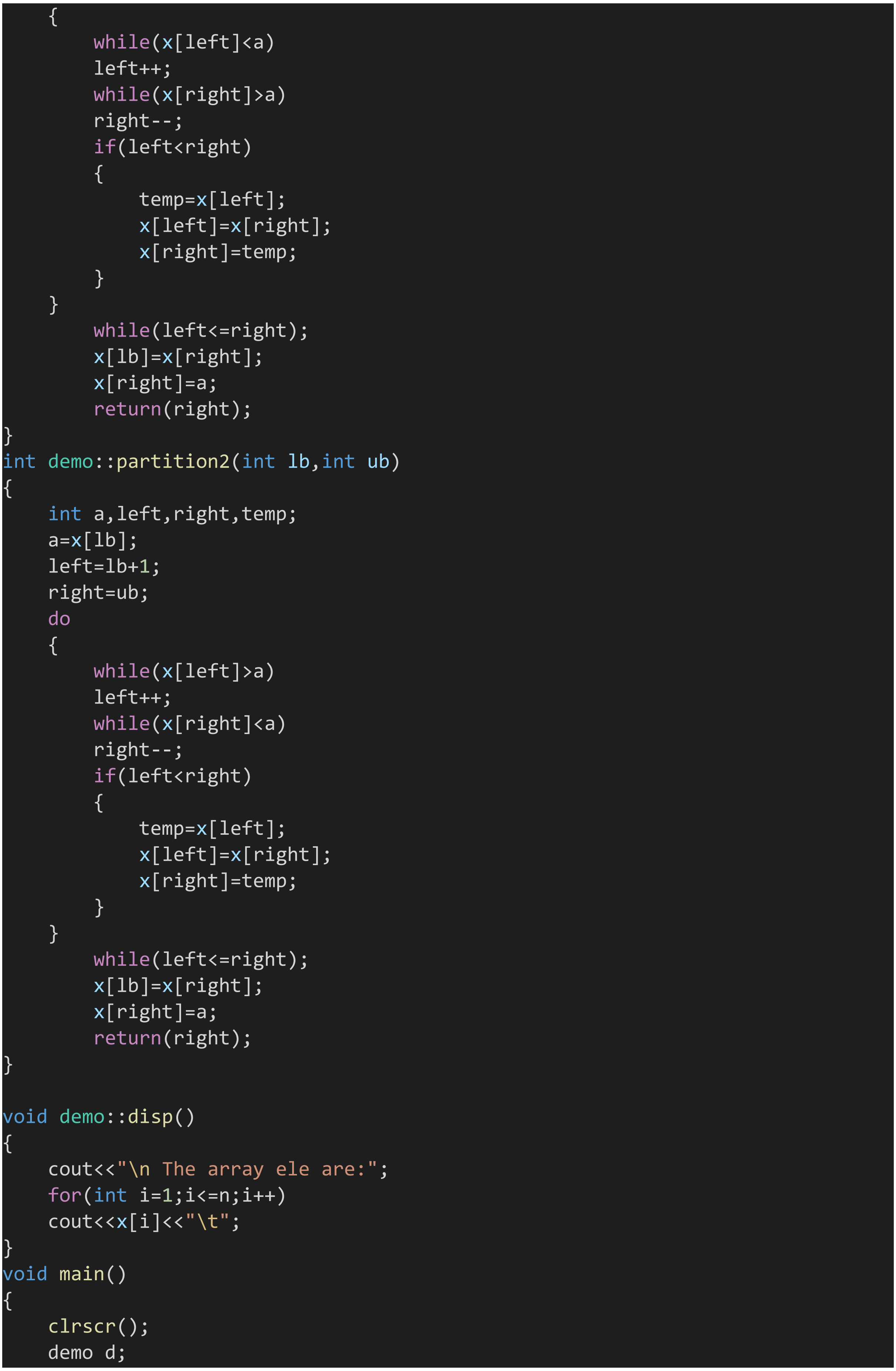


```
while((i<=mid)&&(j<=high))
{
if(a[i]<=a[j])
{
b[k]=a[i];
i++;
k++;
}
else
{
b[k]=a[j];
j++;
k++;
}
}
if(i>mid)
{
while(j<=high)
{
b[k]=a[j];
j++;
k++;
}
}
else
{
while(i<=mid)
{
b[k]=a[i];
i++;
k++;
}
}
for(int k1=low;k1<=high;k1++)
a[k1]=b[k1];
}
void merge::disp()
{
for(int i=1;i<=n;i++)
cout<<a[i]<<"\t";
}
void main()
{
clrscr();
merge m;
m.read();
cout<<"\nAfter sorting\n";
m.disp();
getch();
}
```

9. Assignment Name: Implementation of Quick sort for integer



```
#include<iostream.h>

#include<conio.h>
#include<string.h>
class demo
{
    int x[20],n;
    public:
    void get();
    void asort(int,int);
    void dsort(int,int);
    int partition(int,int);
    int partition2(int,int);
    void disp();
};
void demo::get()
{
    cout<<"\n Enter the array size:";
    cin>>n;
    cout<<"\n Enter the array ele:";
    for(int i=1;i<=n;i++)
    cin>>x[i];
    asort(1,n);
    dsort(1,n);
}
void demo::asort(int p,int q)
{
    if(p<q)
    {
        int j=partition(p,q);
        asort(p,j-1);
        asort(j+1,q);
    }
}
void demo::dsort(int p,int q)
{
    if(p<q)
    {
        int k=partition2(p,q);
        asort(p,k-1);
        asort(k+1,q);
    }
}

int demo::partition(int lb,int ub)
{
    int a,left,right,temp;
    a=x[lb];
    left=lb+1;
    right=ub;
    do
```



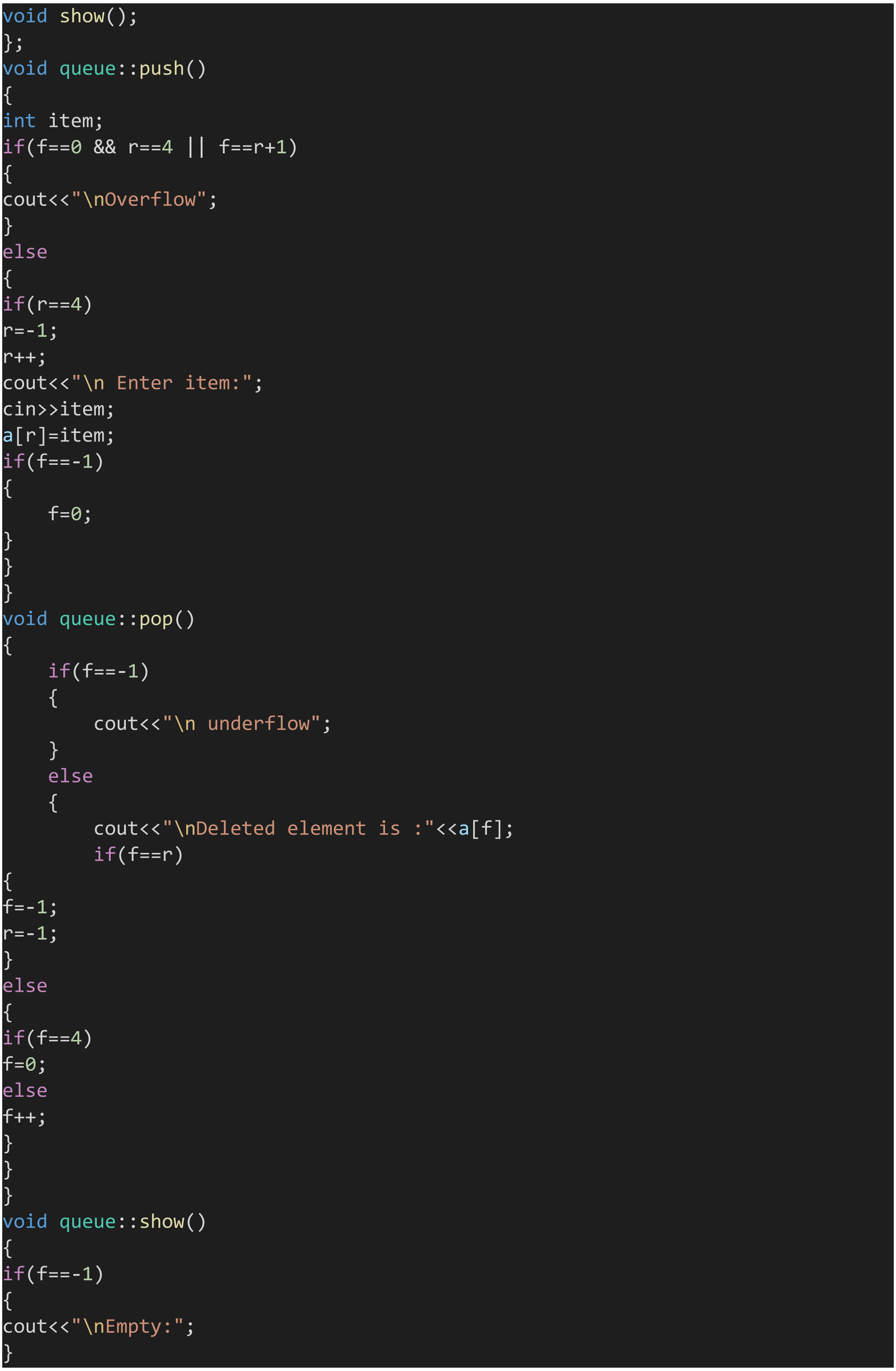
```
    {
        while(x[left]<a)
        left++;
        while(x[right]>a)
        right--;
        if(left<right)
        {
            temp=x[left];
            x[left]=x[right];
            x[right]=temp;
        }
    }
        while(left<=right);
        x[lb]=x[right];
        x[right]=a;
        return(right);
}
int demo::partition2(int lb,int ub)
{
    int a,left,right,temp;
    a=x[lb];
    left=lb+1;
    right=ub;
    do
    {
        while(x[left]>a)
        left++;
        while(x[right]<a)
        right--;
        if(left<right)
        {
            temp=x[left];
            x[left]=x[right];
            x[right]=temp;
        }
    }
        while(left<=right);
        x[lb]=x[right];
        x[right]=a;
        return(right);
}

void demo::disp()
{
    cout<<"\n The array ele are:";
    for(int i=1;i<=n;i++)
    cout<<x[i]<<"\t";
}
void main()
{
    clrscr();
    demo d;
    d.get();
    cout<<"\n After ascending sort";
    d.disp();
    getch();
}
```

10.Assignment Name: Implement linear queue for integer

```
#include<iostream.h>

#include<conio.h>
#include<process.h>
class queue
{
int f,r,q[10],n,i;
public:
queue()
{
f=r=0;
}
void insert();
void del();
void dis();
};
void queue::insert()
{
if(r==3)
cout<<"\nOverflow";
else
{
cout<<"\nEnter n";
cin>>n;
if(f==0)
f=1;
r++;
q[r]=n;
}
}
void queue::del()
{
if(f==0)
{
cout<<"\nUnderflow";
return;
}
else
{
int n;
n=q[f];
if(f==r)
f=r=0;
else
f++;
cout<<"\nDeleted elements is"<<n;
}
}
void queue::dis()
{
if(f==0)
cout<<"\nUnderflow";
else
{
cout<<"\nElements in queue are:";
for(i=f;i<=r;i++)
cout<<q[i]<<"\t";
}
}
void main()
{
clrscr();
queue q;
int ch;
cout<<"\n 1.insert 2.display 3.delete 4.exit \n";
while(ch!=4)
{
cout<<"\Enter*ch:";
cin>>ch;
switch(ch)
{
case 1:q.insert(); break;
case 2:q.dis(); break;
case 3:q.del(); break;
case 4:exit(0);
}
}
getch();
}
```

11.Assignment Name: Implement Circular Queue for integer

```
#include<iostream.h>

#include<conio.h>
#include<process.h>
class queue
{
int a[5],r,f;
public:
queue()
{
f=r=-1;
}
void push();
void pop();
```

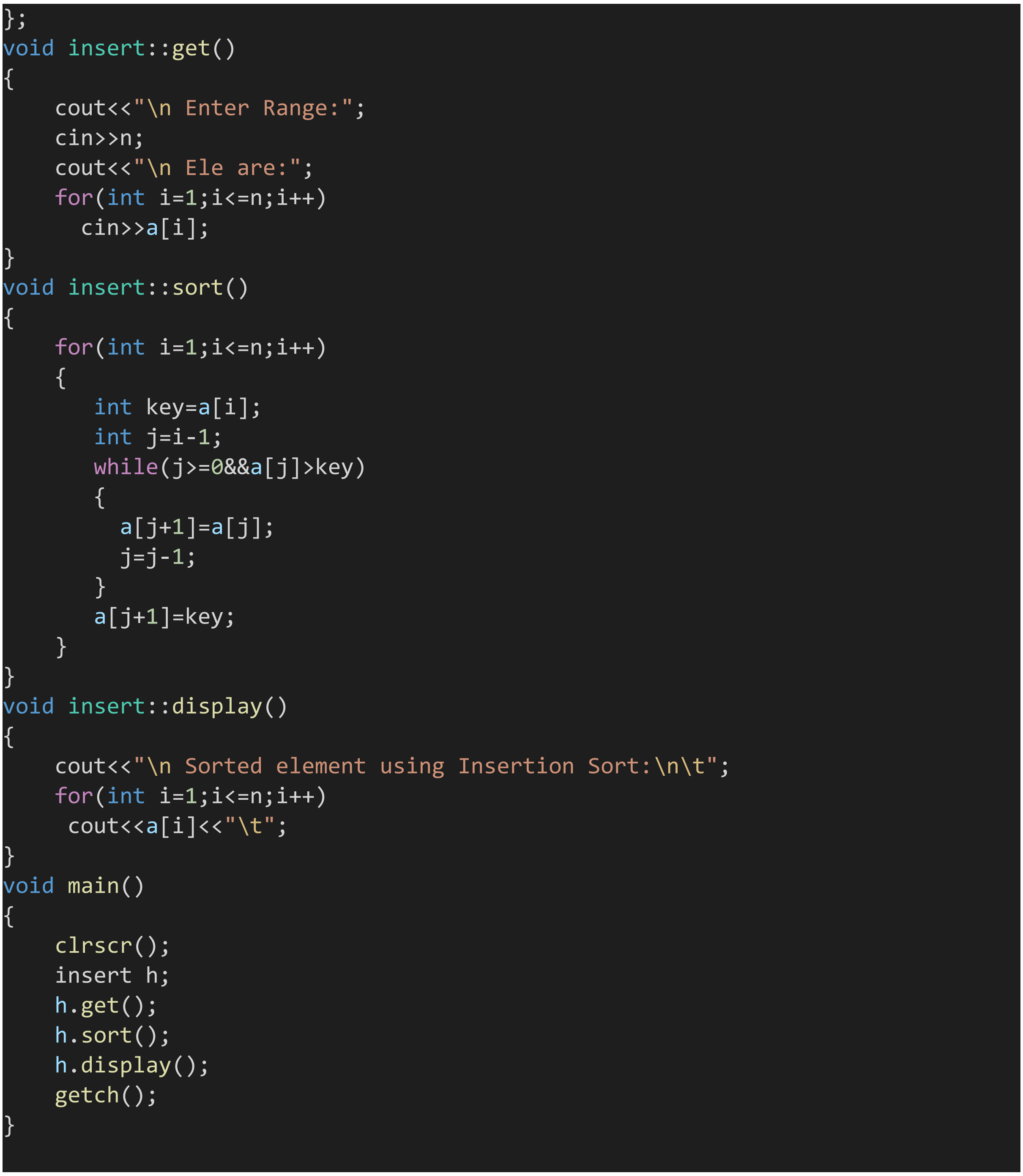


```
void show();
};
void queue::push()
{
int item;
if(f==0 && r==4 || f==r+1)
{
cout<<"\nOverflow";
}
else
{
if(r==4)
r=-1;
r++;
cout<<"\n Enter item:";
cin>>item;
a[r]=item;
if(f==-1)
{
    f=0;
}
}
}
void queue::pop()
{
    if(f==-1)
    {
        cout<<"\n underflow";
    }
    else
    {
        cout<<"\nDeleted element is :"<<a[f];
        if(f==r)
{
f=-1;
r=-1;
}
else
{
if(f==4)
f=0;
else
f++;
}
}
}
void queue::show()
{
if(f==-1)
{
cout<<"\nEmpty:";
}
```

```
else if(f<=r)
{
for(int i=f;i<r;i++)
{
cout<<"\n"<<a[i];
}
}
else
{
for(int i=f;i<=4;i++)
{
cout<<"\n"<<a[i];
}
for(int j=0;j<=r;j++)
{
cout<<"\n"<<a[i];
}
}
}
void main()
{
queue s;
int ch;
clrscr();
do
{
cout<<"\n 1:Push 2:Pop 3:Show 4:exit ";
cout<<"\nEnter choice";
cin>>ch;
switch(ch)
{
case 1: s.push();break;
case 2: s.pop();break;
case 3: s.show();break;
default: cout<<"\n Wrong Choice";
}
}while(ch<=3);
}
```

12.Assignment Name: Implementation of Insertion Sort

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<math.h>
class insert
{
    int n,a[10];
public:
    void get();
    void sort();
    void display();
```



```
};
void insert::get()
{
    cout<<"\n Enter Range:";
    cin>>n;
    cout<<"\n Ele are:";
    for(int i=1;i<=n;i++)
      cin>>a[i];
}
void insert::sort()
{
    for(int i=1;i<=n;i++)
    {
       int key=a[i];
       int j=i-1;
       while(j>=0&&a[j]>key)
       {
         a[j+1]=a[j];
         j=j-1;
       }
       a[j+1]=key;
    }
}
void insert::display()
{
    cout<<"\n Sorted element using Insertion Sort:\n\t";
    for(int i=1;i<=n;i++)
     cout<<a[i]<<"\t";
}
void main()
{
    clrscr();
    insert h;
    h.get();
    h.sort();
    h.display();
    getch();
}
```