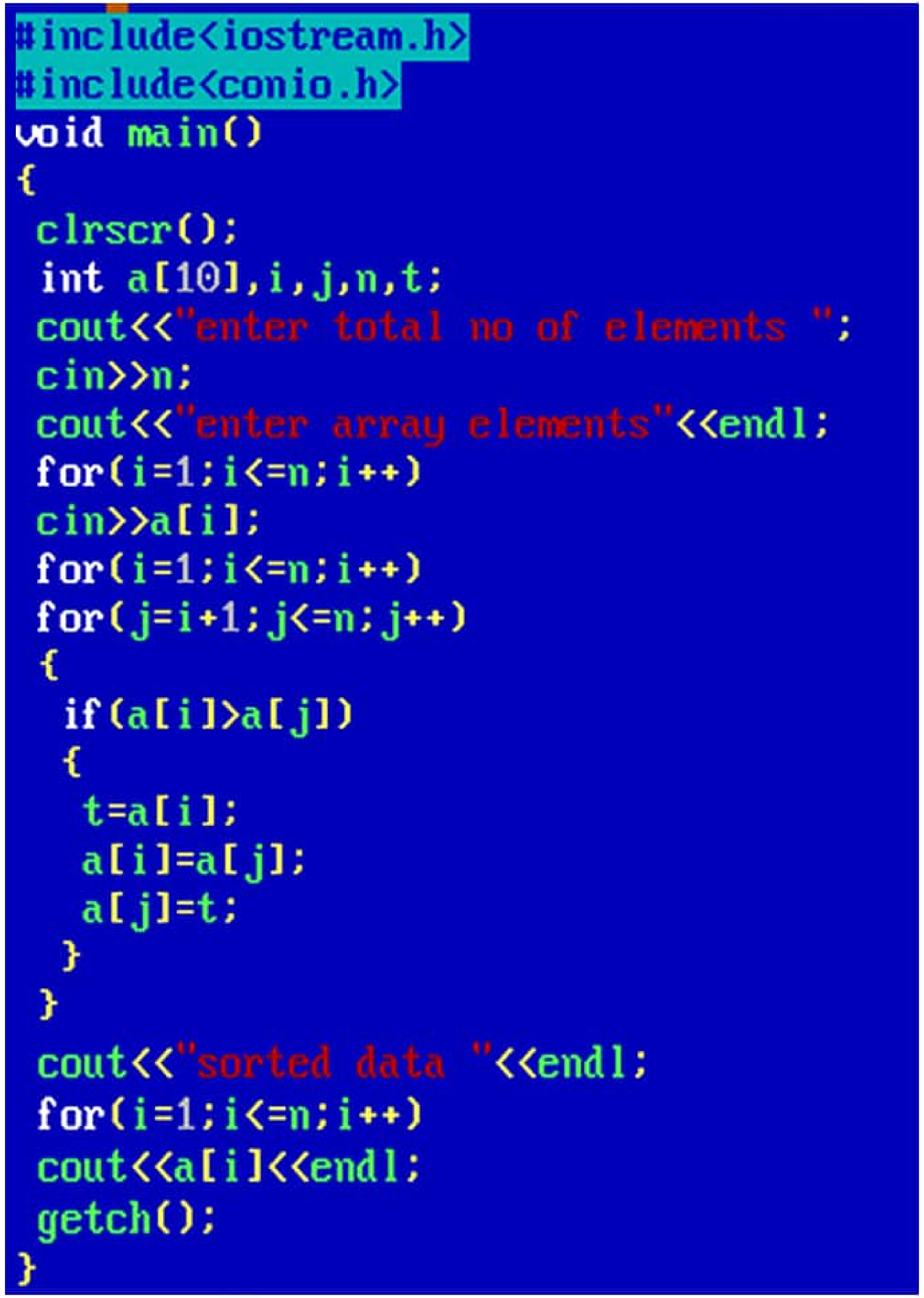
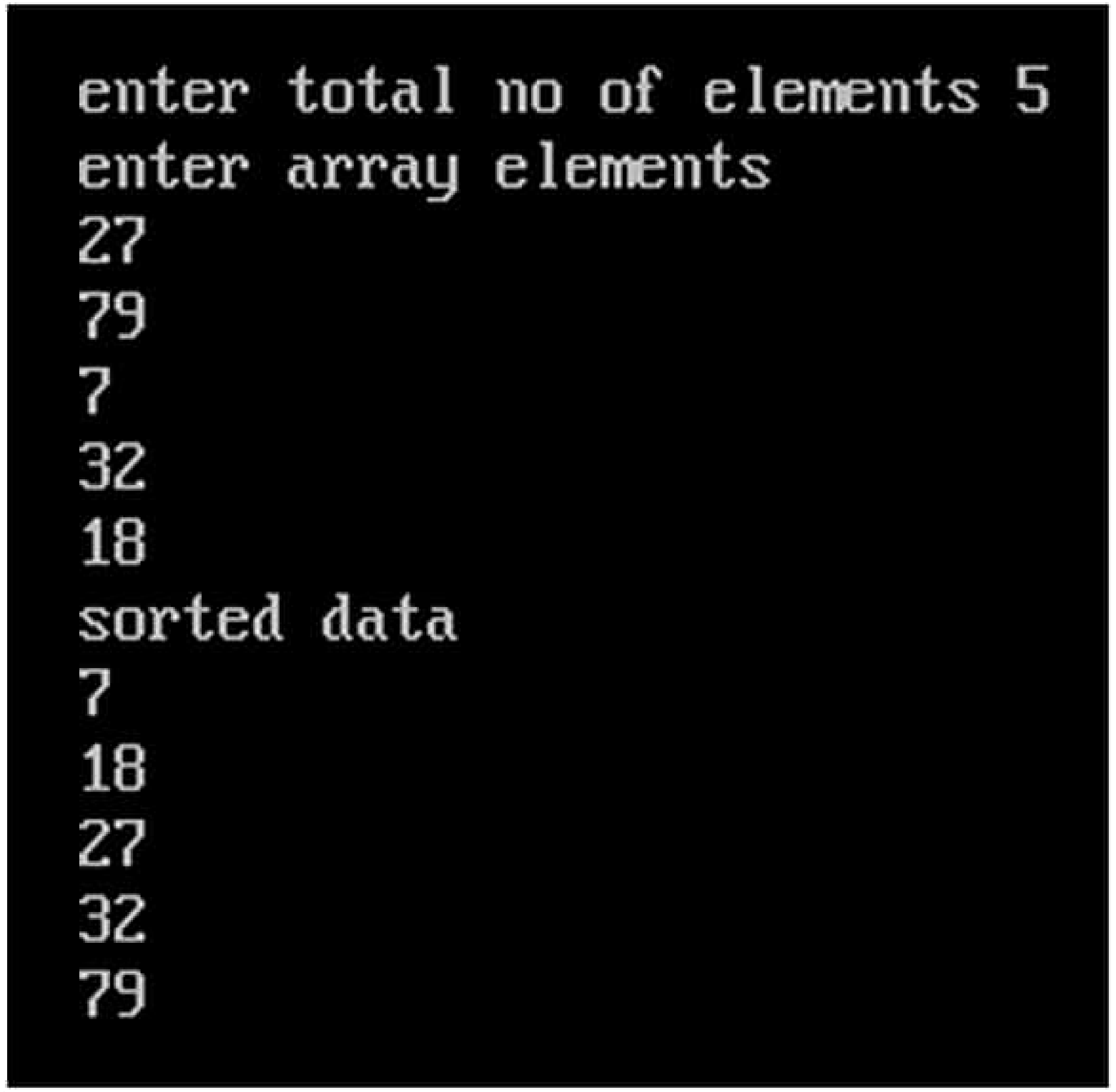
sorted

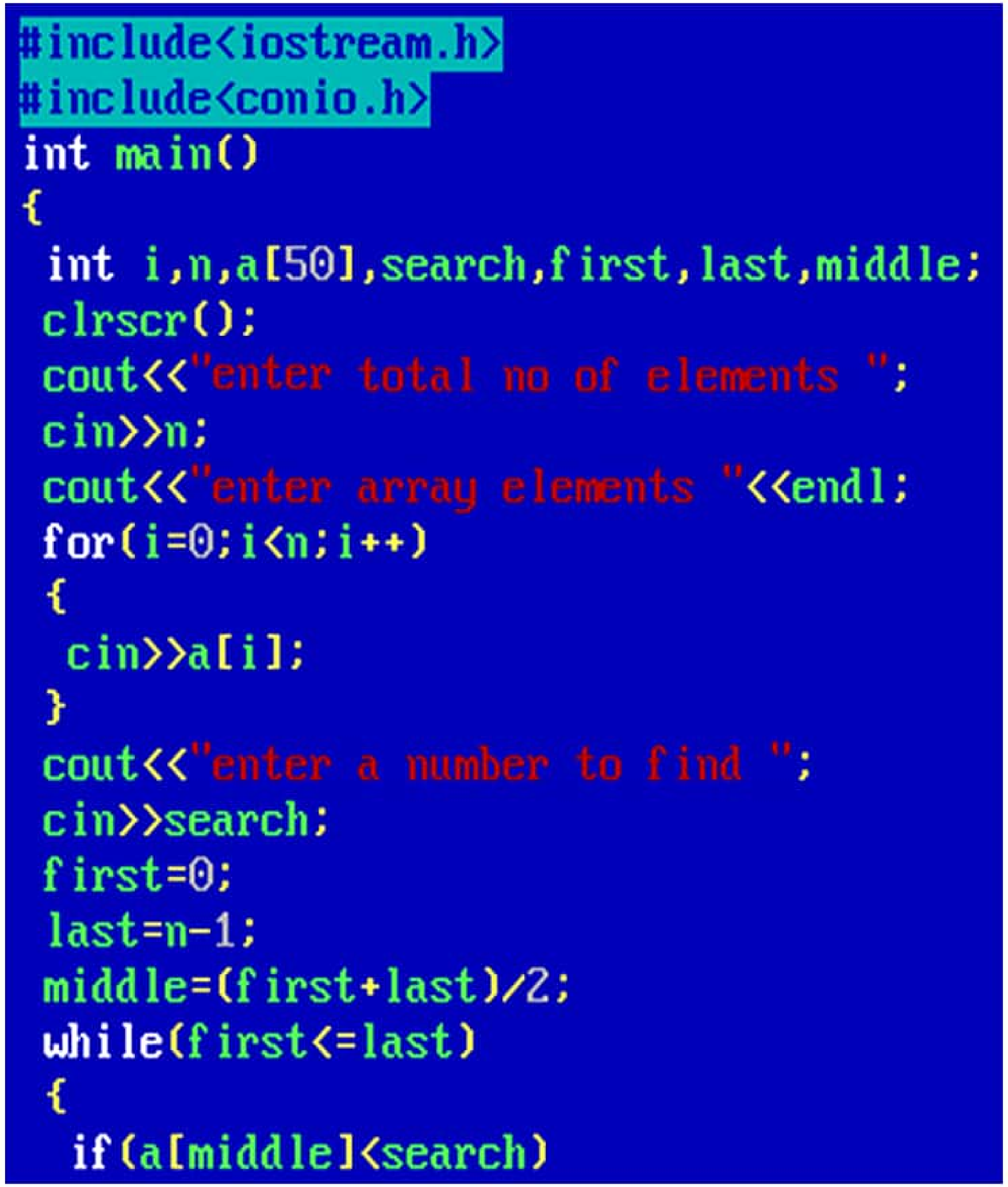


```
#include<iostream.h>
#include<conio.h>
void main()
{
 clrscr();
 int a[10],i,j,n,t;
 cout<<"enter total no of elements ";
 cin>>n;
 cout<<"enter array elements"<<endl;
 for(i=1;i<=n;i++)
 cin>>a[i];
 for(i=1;i<=n;i++)
 for(j=i+1;j<=n;j++)
 {
  if(a[i]>a[j])
  {
   t=a[i];
   a[i]=a[j];
   a[j]=t;
  }
 }
 cout<<"sorted data "<<endl;
 for(i=1;i<=n;i++)
 cout<<a[i]<<endl;
 getch();
}
```

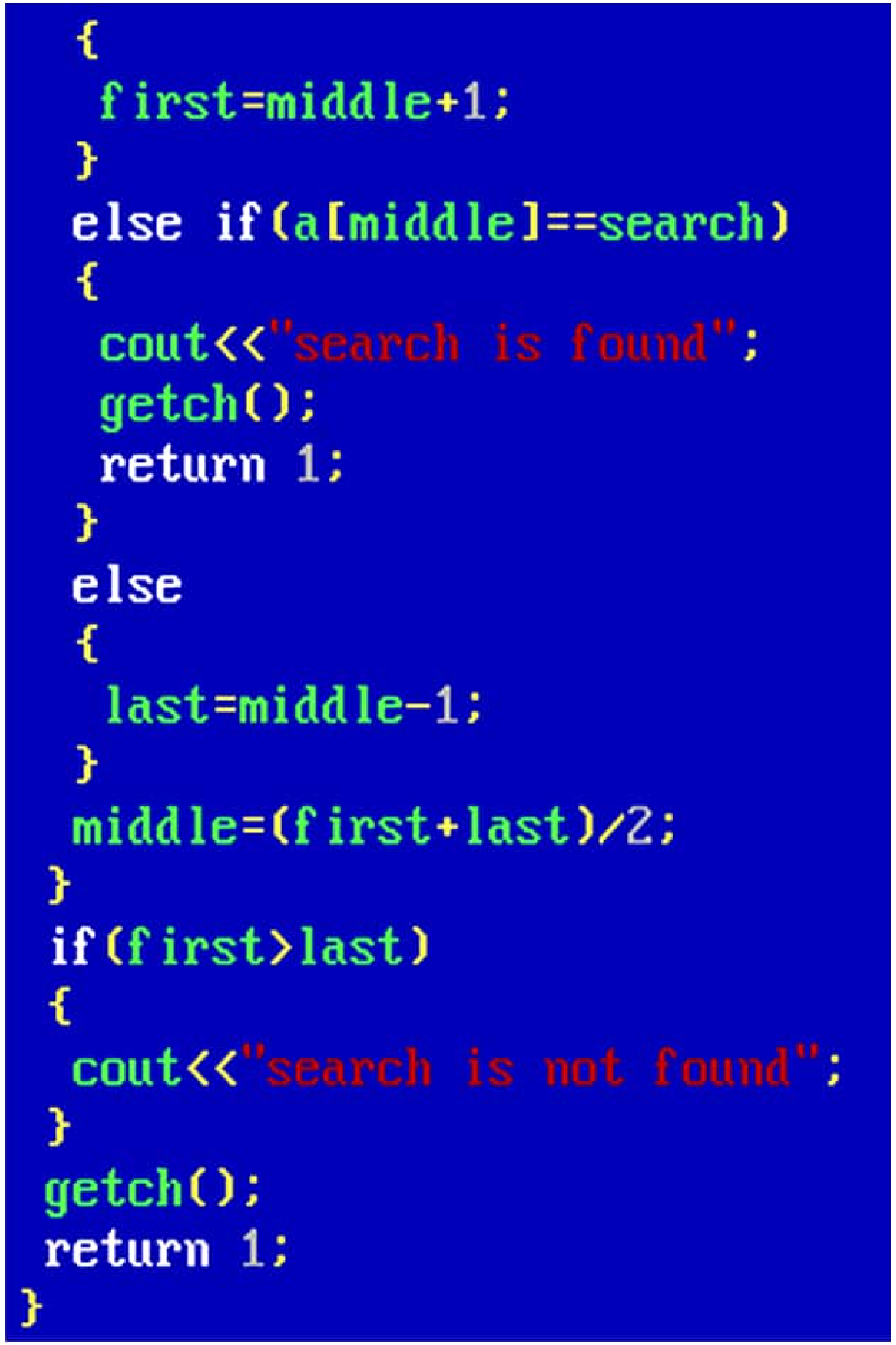


```
enter total no of elements 5
enter array elements
27
79
7
32
18
sorted data
7
18
27
32
79
```

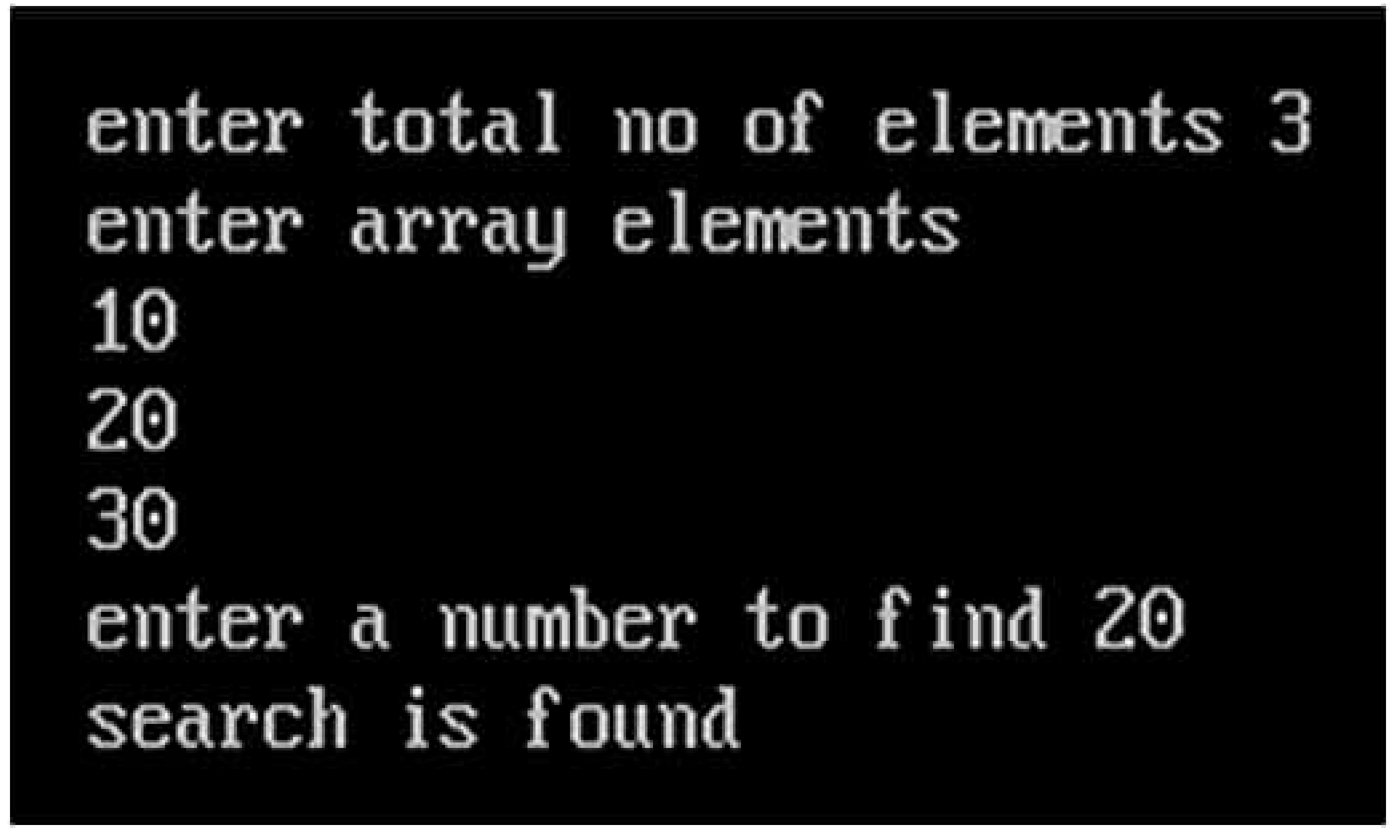
binary search



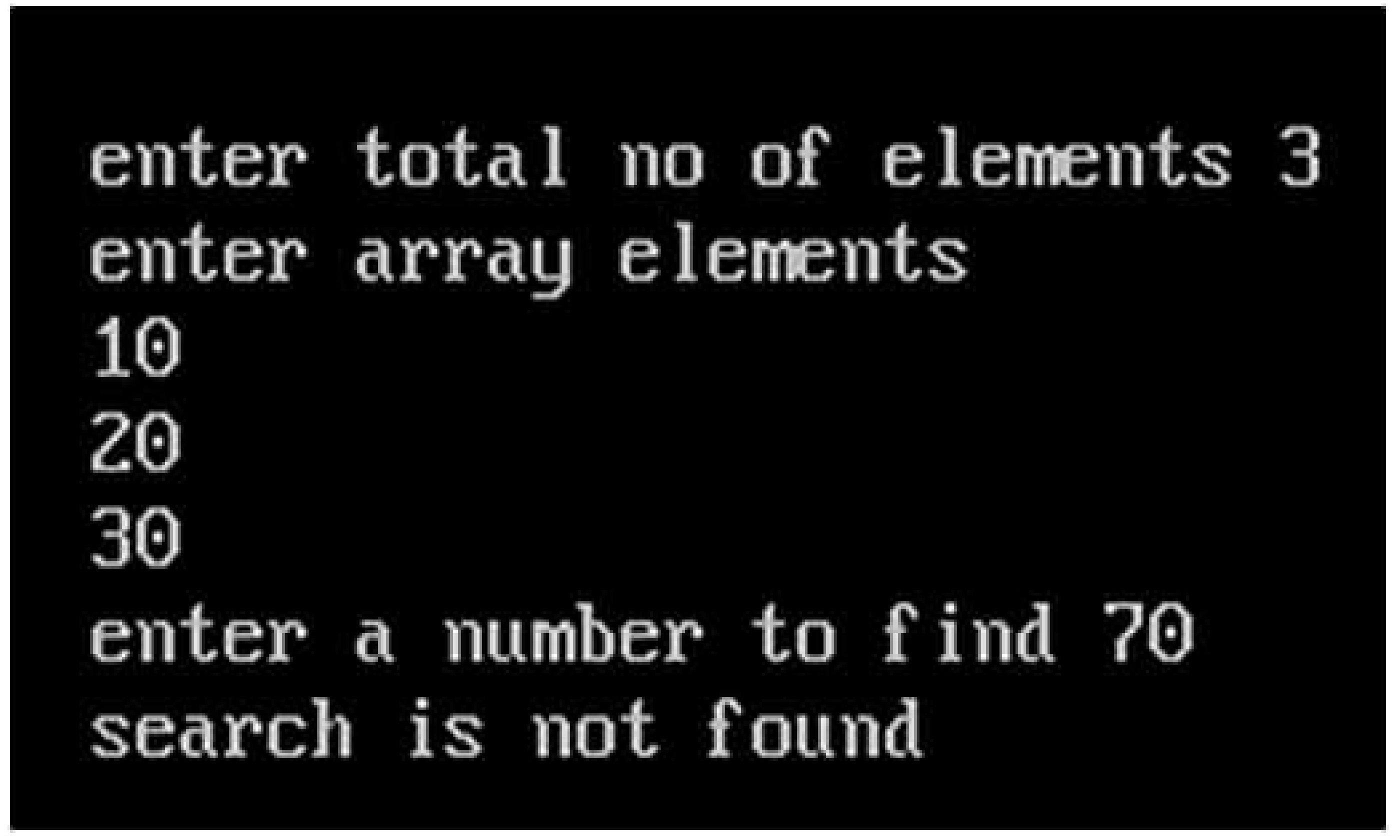
```
#include<iostream.h>
#include<conio.h>
int main()
{
 int i,n,a[50],search,first,last,middle;
 clrscr();
 cout<<"enter total no of elements ";
 cin>>n;
 cout<<"enter array elements "<<endl;
 for(i=0;i<n;i++)
 {
  cin>>a[i];
 }
 cout<<"enter a number to find ";
 cin>>search;
 first=0;
 last=n-1;
 middle=(first+last)/2;
 while(first<=last)
 {
  if(a[middle]<search)
```



```
  {
   first=middle+1;
  }
  else if(a[middle]==search)
  {
   cout<<"search is found";
   getch();
   return 1;
  }
  else
  {
   last=middle-1;
  }
  middle=(first+last)/2;
 }
 if(first>last)
 {
  cout<<"search is not found";
 }
 getch();
 return 1;
}
```

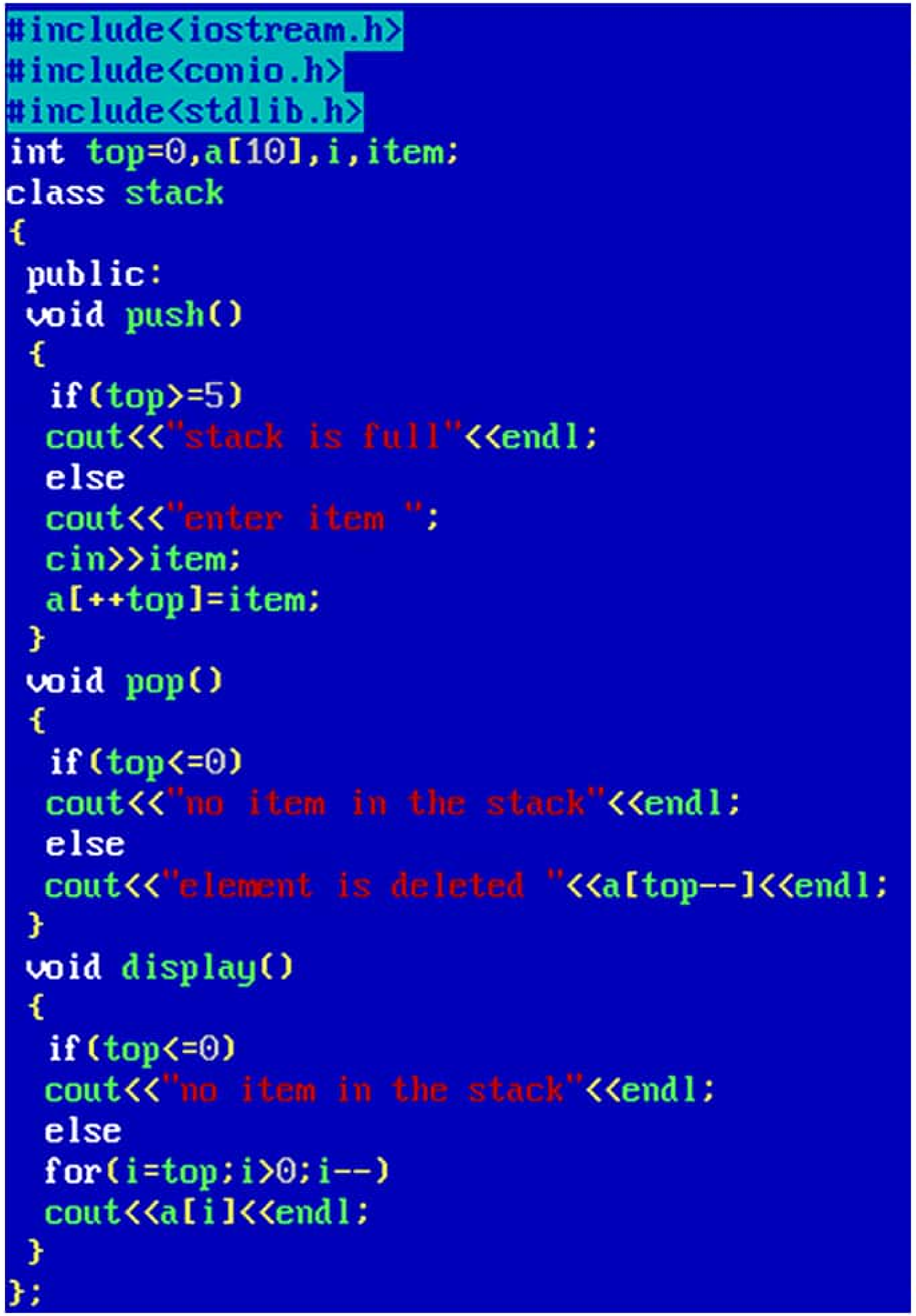


```
enter total no of elements 3
enter array elements
10
20
30
enter a number to find 20
search is found
```

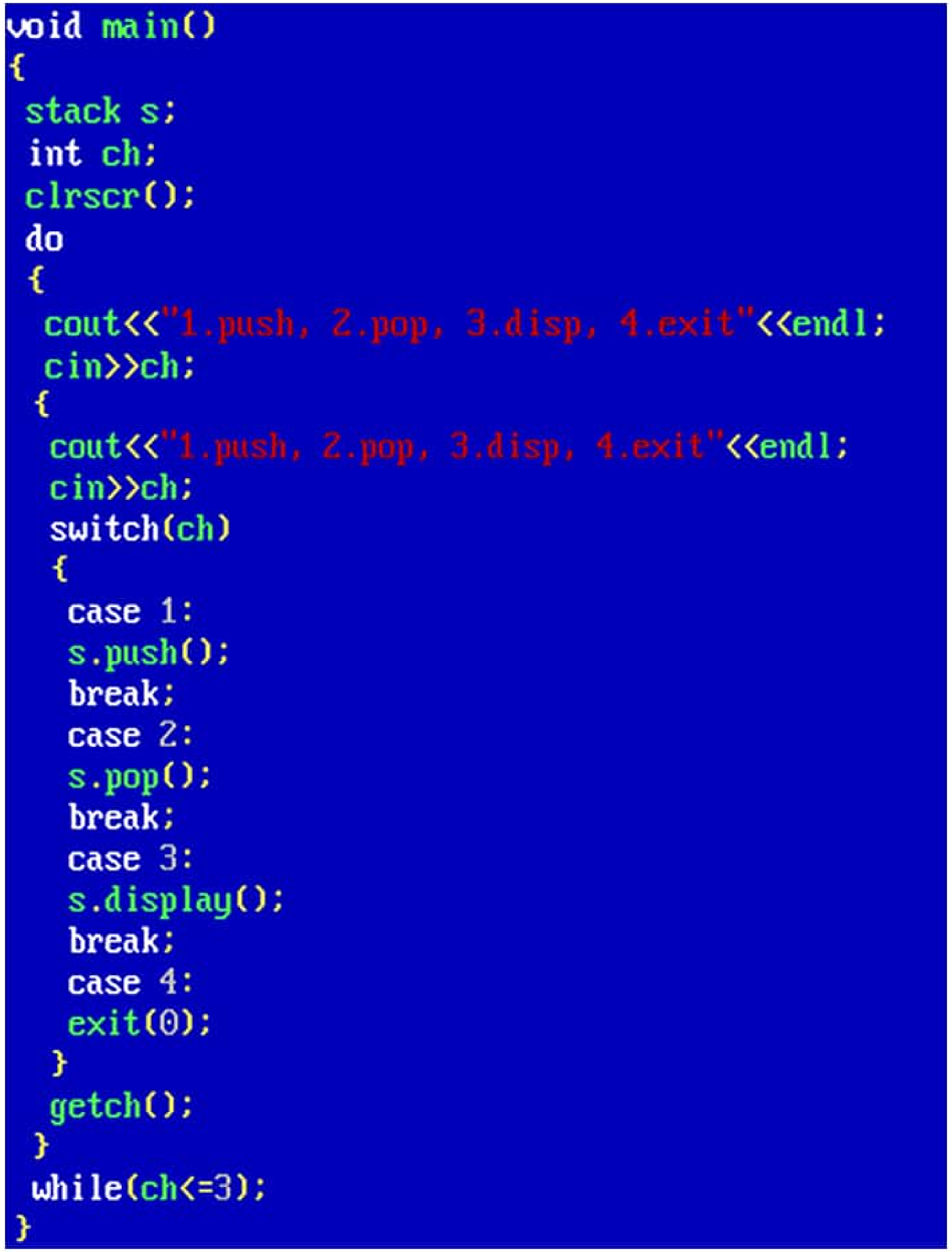


```
enter total no of elements 3
enter array elements
10
20
30
enter a number to find 70
search is not found
```

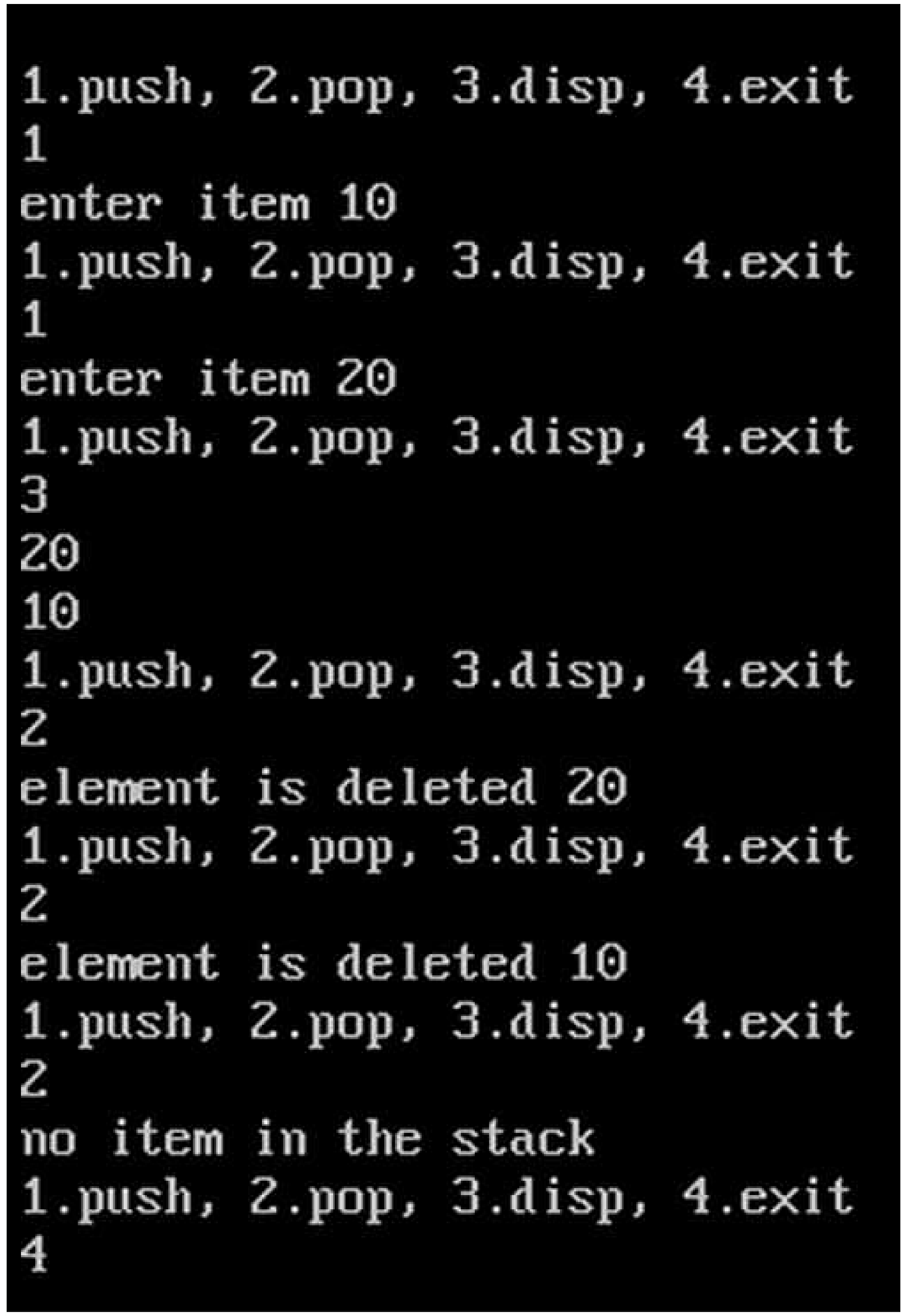
stack using array



```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
int top=0,a[10],i,item;
class stack
{
 public:
 void push()
 {
  if(top>=5)
  cout<<"stack is full"<<endl;
  else
  cout<<"enter item ";
  cin>>item;
  a[++top]=item;
 }
 void pop()
 {
  if(top<=0)
  cout<<"no item in the stack"<<endl;
  else
  cout<<"element is deleted "<<a[top--]<<endl;
 }
 void display()
 {
  if(top<=0)
  cout<<"no item in the stack"<<endl;
  else
  for(i=top;i>0;i--)
  cout<<a[i]<<endl;
 }
};
```

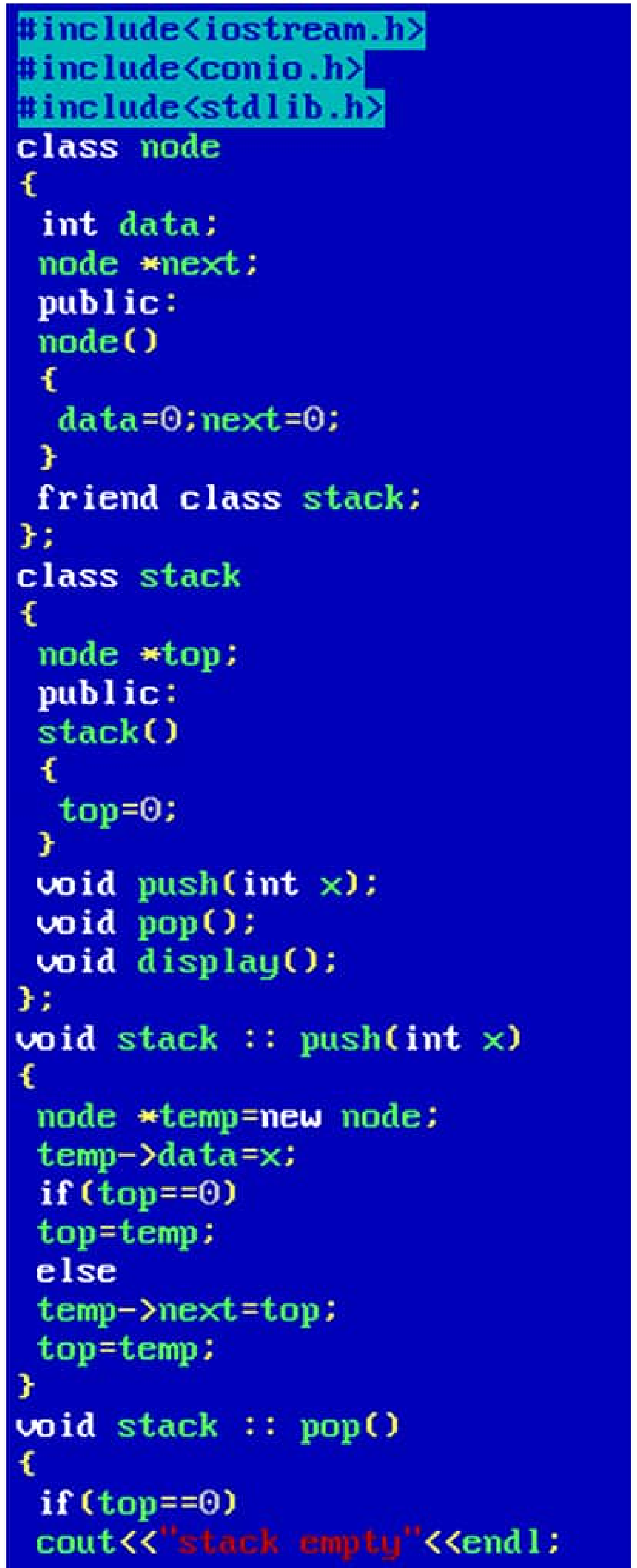


```
void main()
{
 stack s;
 int ch;
 clrscr();
 do
 {
  cout<<"1.push, 2.pop, 3.disp, 4.exit"<<endl;
  cin>>ch;
  {
   cout<<"1.push, 2.pop, 3.disp, 4.exit"<<endl;
   cin>>ch;
   switch(ch)
   {
    case 1:
    s.push();
    break;
    case 2:
    s.pop();
    break;
    case 3:
    s.display();
    break;
    case 4:
    exit(0);
   }
   getch();
 }
 while(ch<=3);
}
```

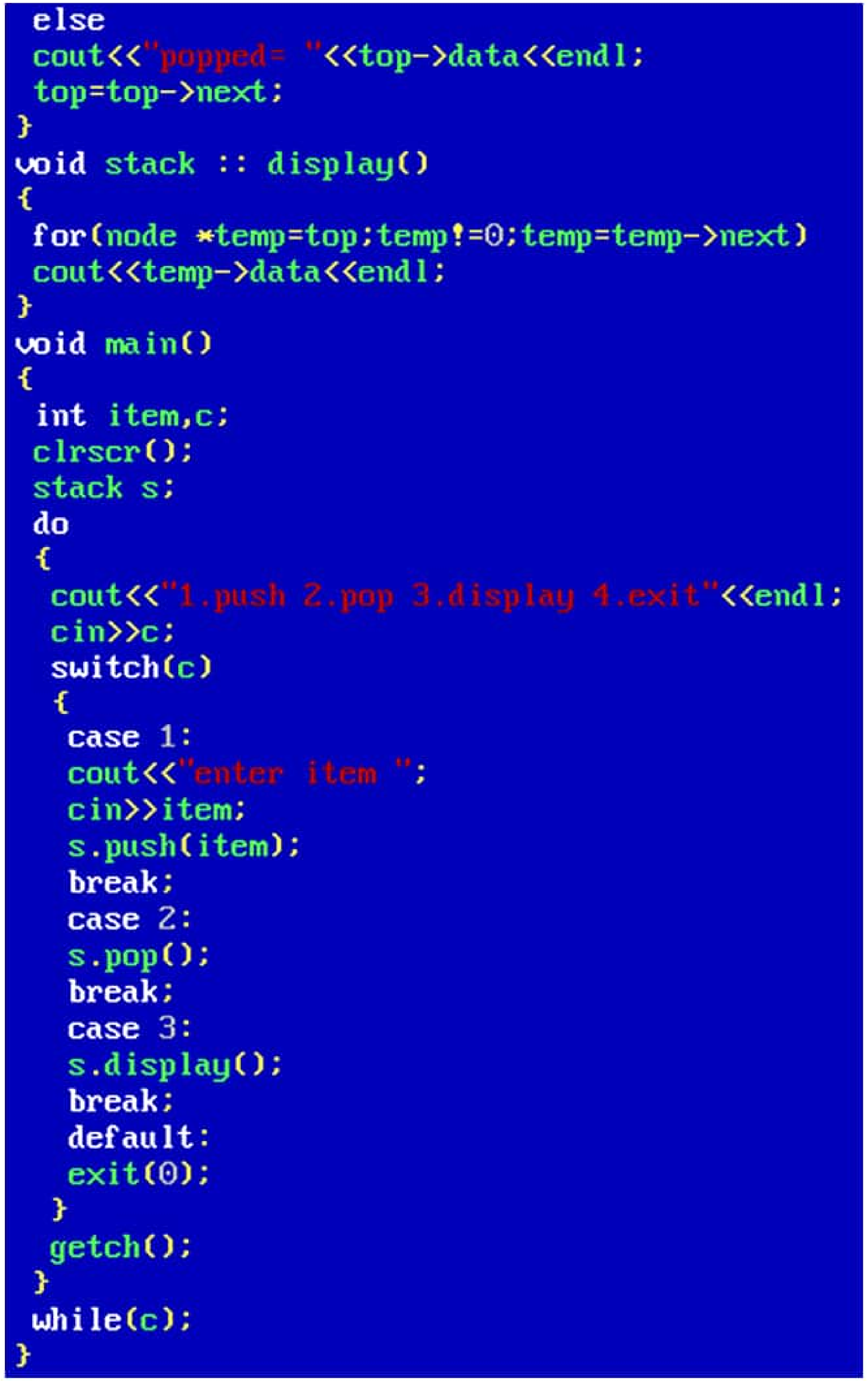


```
1.push, 2.pop, 3.disp, 4.exit
1
enter item 10
1.push, 2.pop, 3.disp, 4.exit
1
enter item 20
1.push, 2.pop, 3.disp, 4.exit
3
20
10
1.push, 2.pop, 3.disp, 4.exit
2
element is deleted 20
1.push, 2.pop, 3.disp, 4.exit
2
element is deleted 10
1.push, 2.pop, 3.disp, 4.exit
2
no item in the stack
1.push, 2.pop, 3.disp, 4.exit
4
```

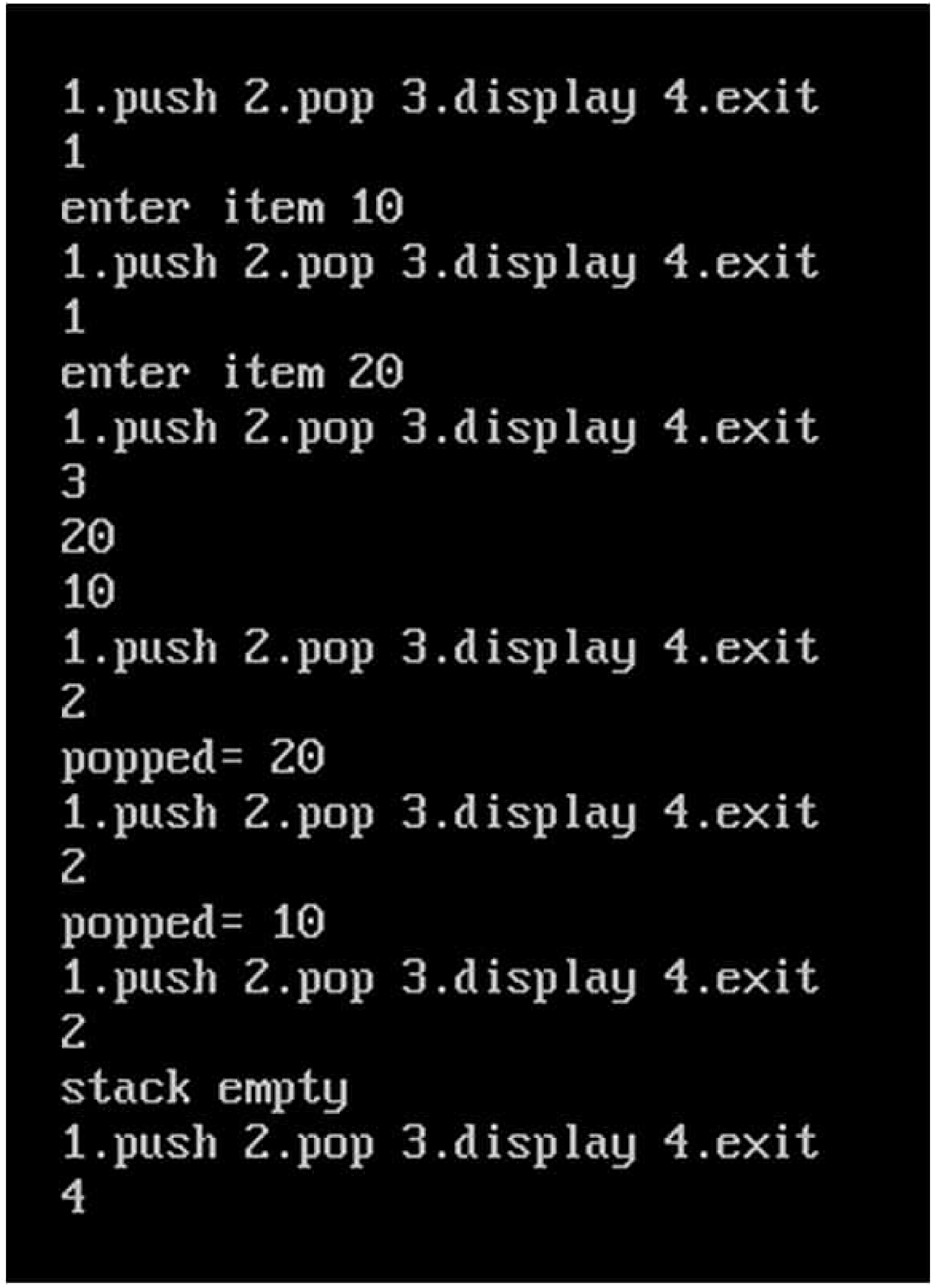
stack pointer



```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class node
{
 int data;
 node *next;
 public:
 node()
 {
  data=0;next=0;
 }
 friend class stack;
};
class stack
{
 node *top;
 public:
 stack()
 {
  top=0;
 }
 void push(int x);
 void pop();
 void display();
};
void stack :: push(int x)
{
 node *temp=new node;
 temp->data=x;
 if(top==0)
 top=temp;
 else
 temp->next=top;
 top=temp;
}
void stack :: pop()
{
 if(top==0)
 cout<<"stack empty"<<endl;
```

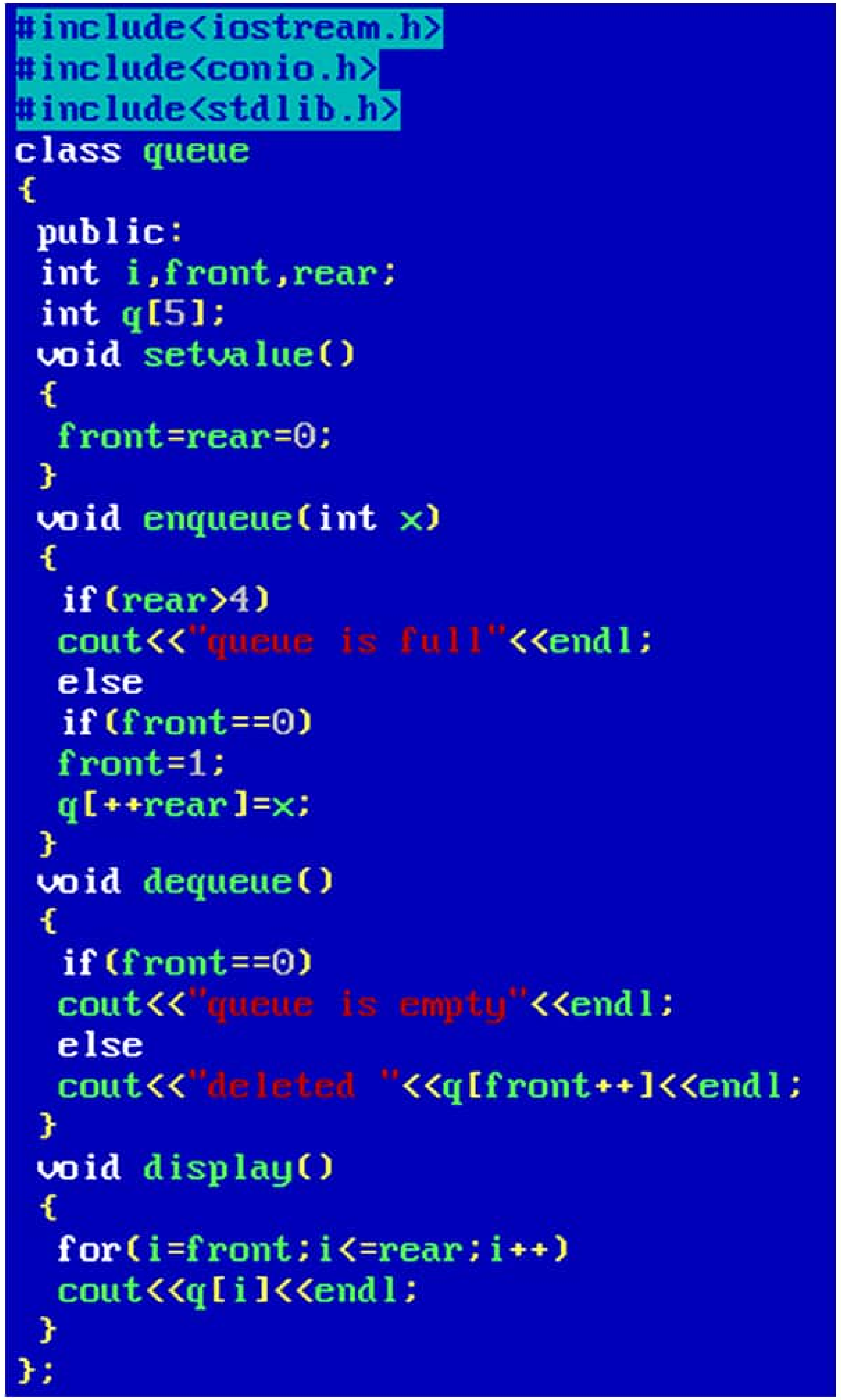


```
 else
 cout<<"popped= "<<top->data<<endl;
 top=top->next;
}
void stack :: display()
{
 for(node *temp=top;temp!=0;temp=temp->next)
 cout<<temp->data<<endl;
}
void main()
{
 int item,c;
 clrscr();
 stack s;
 do
 {
  cout<<"1.push 2.pop 3.display 4.exit"<<endl;
  cin>>c;
  switch(c)
  {
   case 1:
   cout<<"enter item ";
   cin>>item;
   s.push(item);
   break;
   case 2:
   s.pop();
   break;
   case 3:
   s.display();
   break;
   default:
   exit(0);
  }
  getch();
 }
 while(c);
}
```

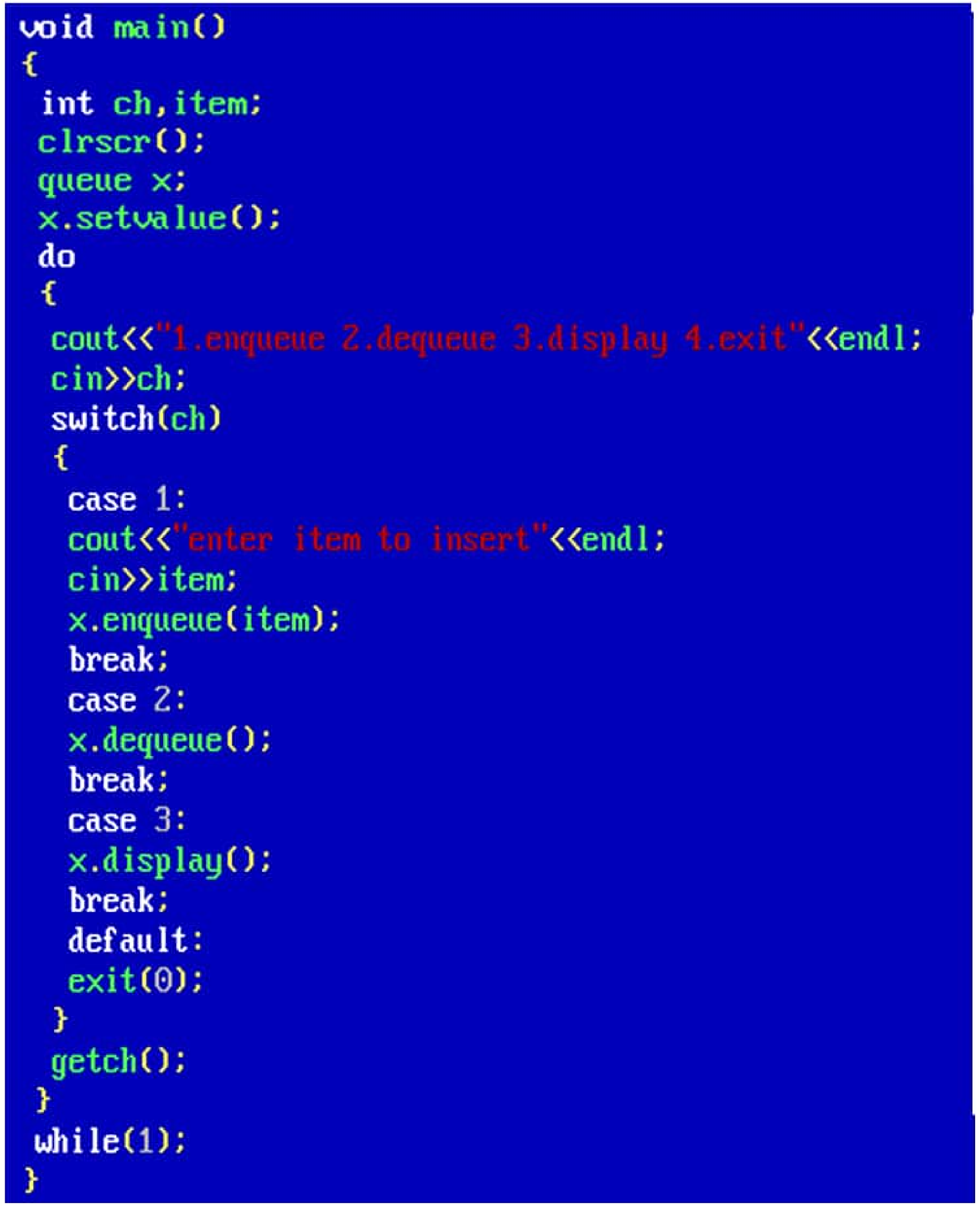


```
1.push 2.pop 3.display 4.exit
1
enter item 10
1.push 2.pop 3.display 4.exit
1
enter item 20
1.push 2.pop 3.display 4.exit
3
20
10
1.push 2.pop 3.display 4.exit
2
popped= 20
1.push 2.pop 3.display 4.exit
2
popped= 10
1.push 2.pop 3.display 4.exit
2
stack empty
1.push 2.pop 3.display 4.exit
4
```

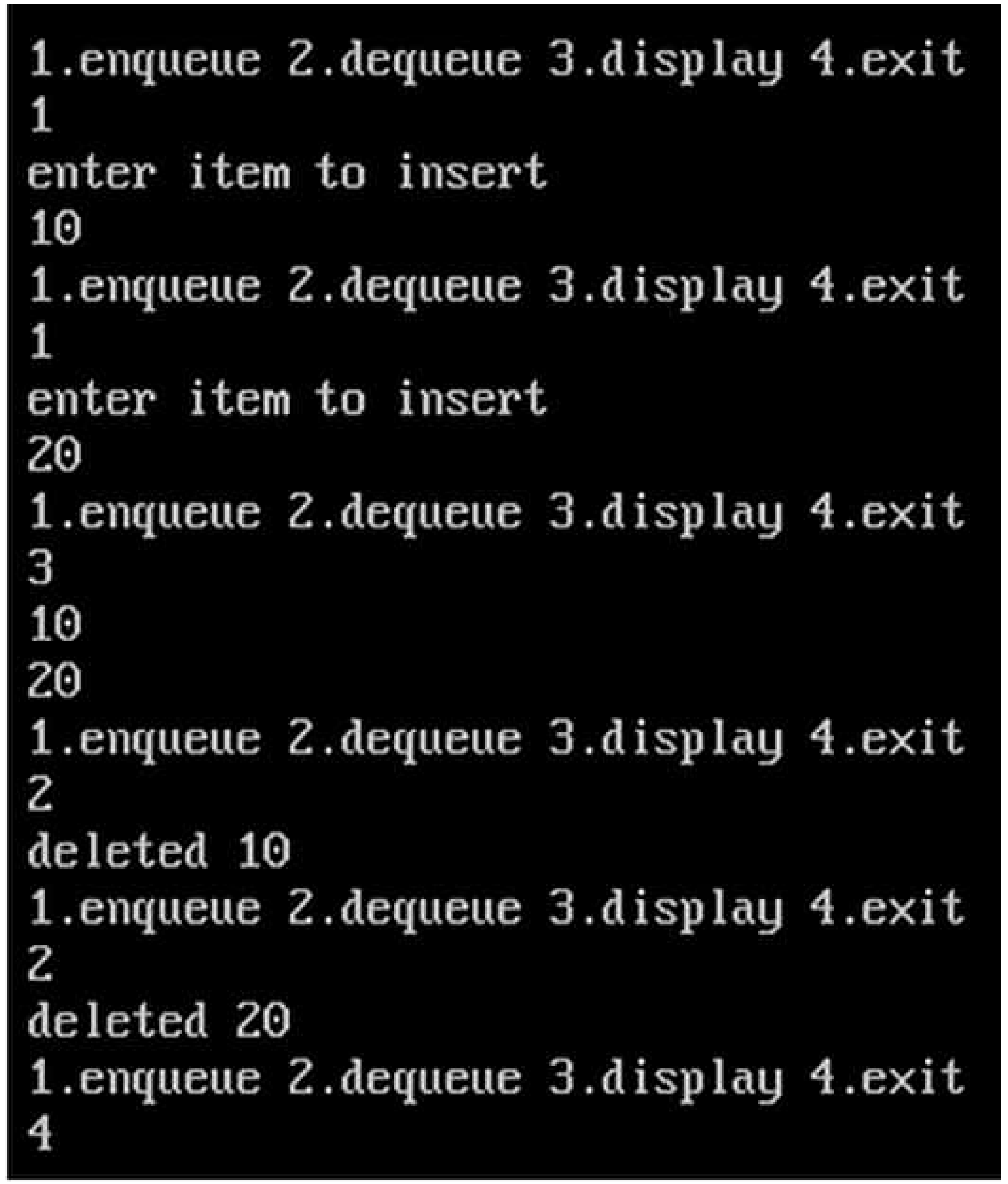
queue array



```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class queue
{
 public:
 int i,front,rear;
 int q[5];
 void setvalue()
 {
  front=rear=0;
 }
 void enqueue(int x)
 {
  if(rear>4)
  cout<<"queue is full"<<endl;
  else
  if(front==0)
  front=1;
  q[++rear]=x;
 }
 void dequeue()
 {
  if(front==0)
  cout<<"queue is empty"<<endl;
  else
  cout<<"deleted "<<q[front++]<<endl;
 }
 void display()
 {
  for(i=front;i<=rear;i++)
  cout<<q[i]<<endl;
 }
};
```

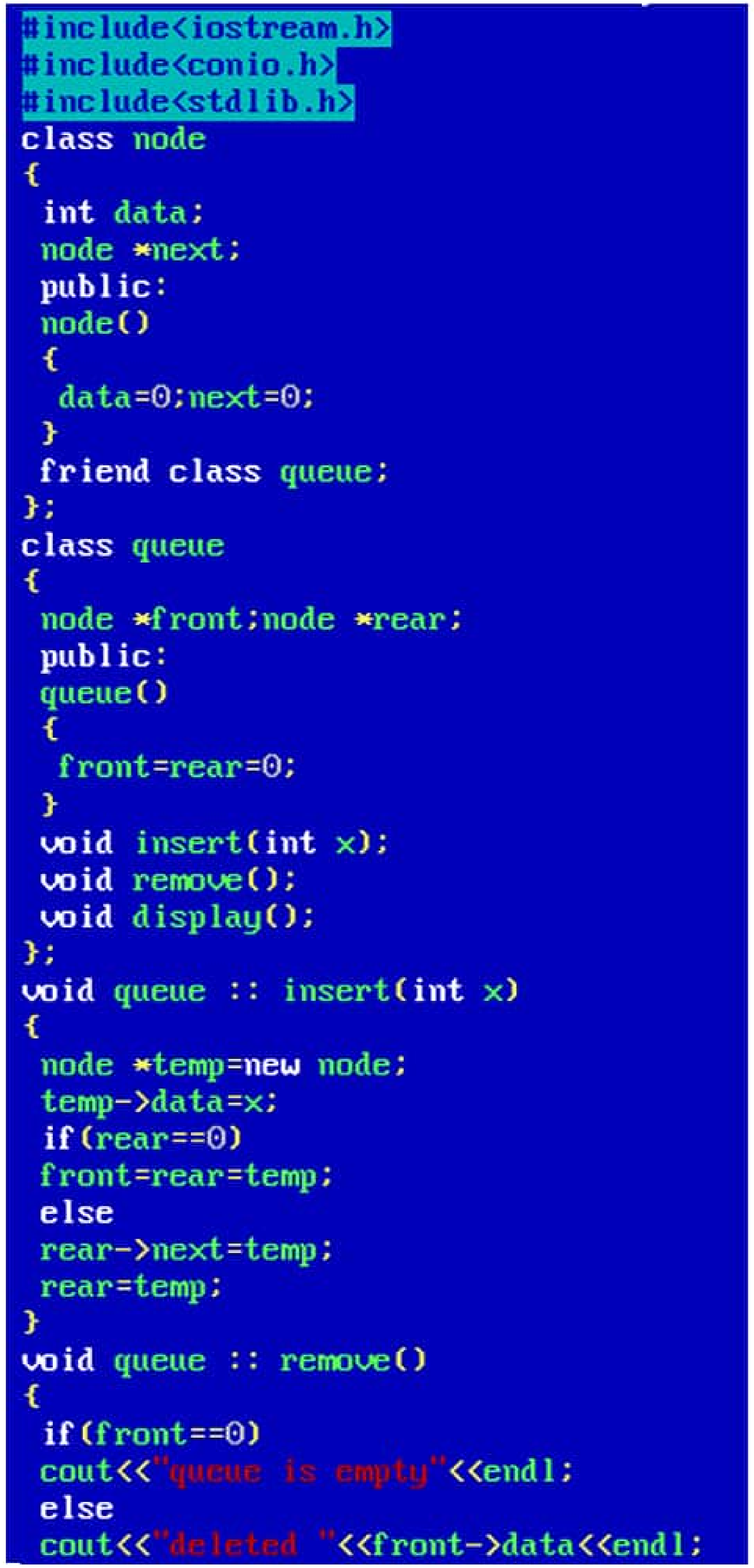


```
void main()
{
 int ch,item;
 clrscr();
 queue x;
 x.setvalue();
 do
 {
  cout<<"1.enqueue 2.dequeue 3.display 4.exit"<<endl;
  cin>>ch;
  switch(ch)
  {
   case 1:
   cout<<"enter item to insert"<<endl;
   cin>>item;
   x.enqueue(item);
   break;
   case 2:
   x.dequeue();
   break;
   case 3:
   x.display();
   break;
   default:
   exit(0);
  }
  getch();
 }
 while(1);
}
```

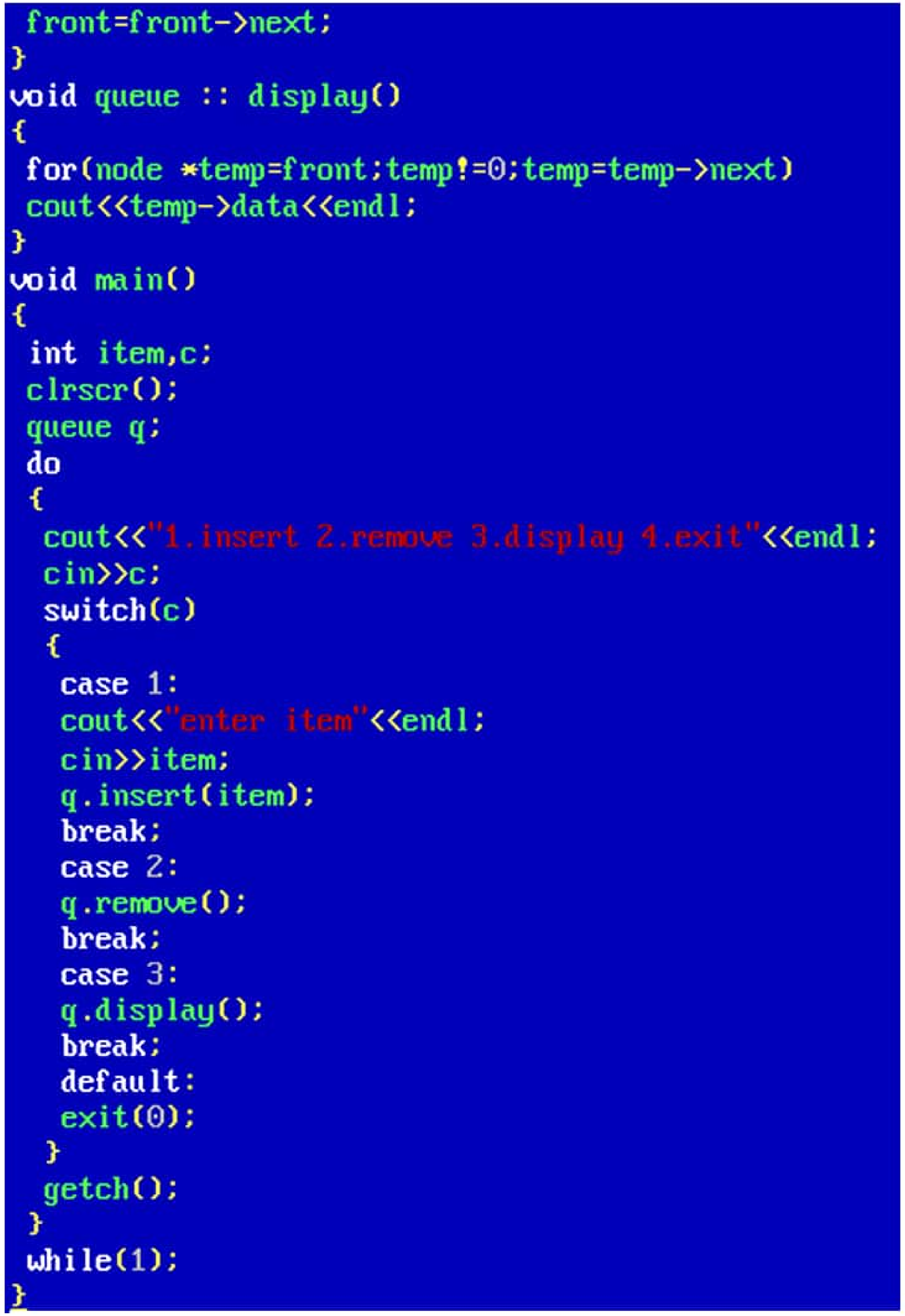


```
1.enqueue 2.dequeue 3.display 4.exit
1
enter item to insert
10
1.enqueue 2.dequeue 3.display 4.exit
1
enter item to insert
20
1.enqueue 2.dequeue 3.display 4.exit
3
10
20
1.enqueue 2.dequeue 3.display 4.exit
2
deleted 10
1.enqueue 2.dequeue 3.display 4.exit
2
deleted 20
1.enqueue 2.dequeue 3.display 4.exit
4
```

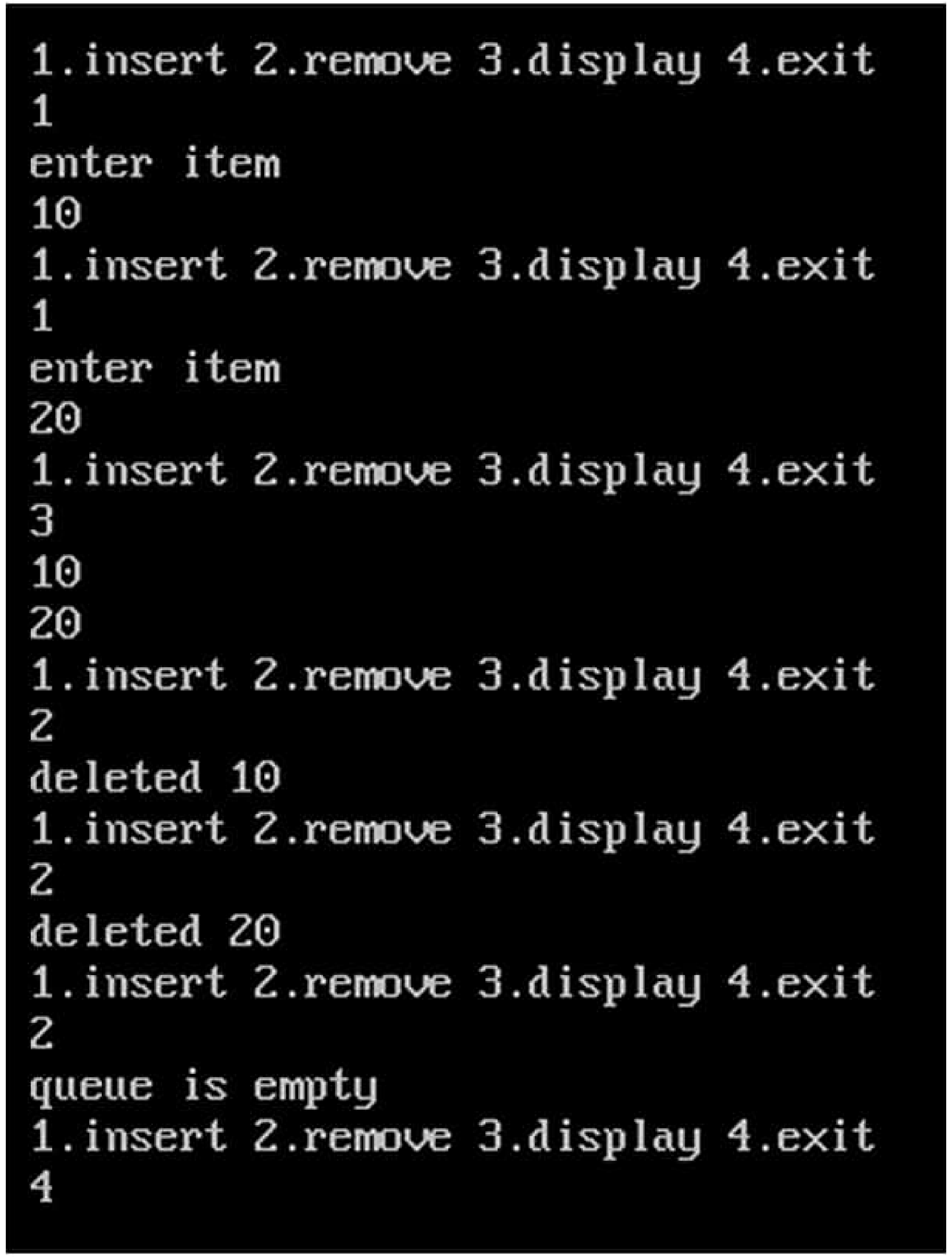
queue pointer



```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
class node
{
 int data;
 node *next;
 public:
 node()
 {
  data=0;next=0;
 }
 friend class queue;
};
class queue
{
 node *front;node *rear;
 public:
 queue()
 {
  front=rear=0;
 }
 void insert(int x);
 void remove();
 void display();
};
void queue :: insert(int x)
{
 node *temp=new node;
 temp->data=x;
 if(rear==0)
 front=rear=temp;
 else
 rear->next=temp;
 rear=temp;
}
void queue :: remove()
{
 if(front==0)
 cout<<"queue is empty"<<endl;
 else
 cout<<"deleted "<<front->data<<endl;
```

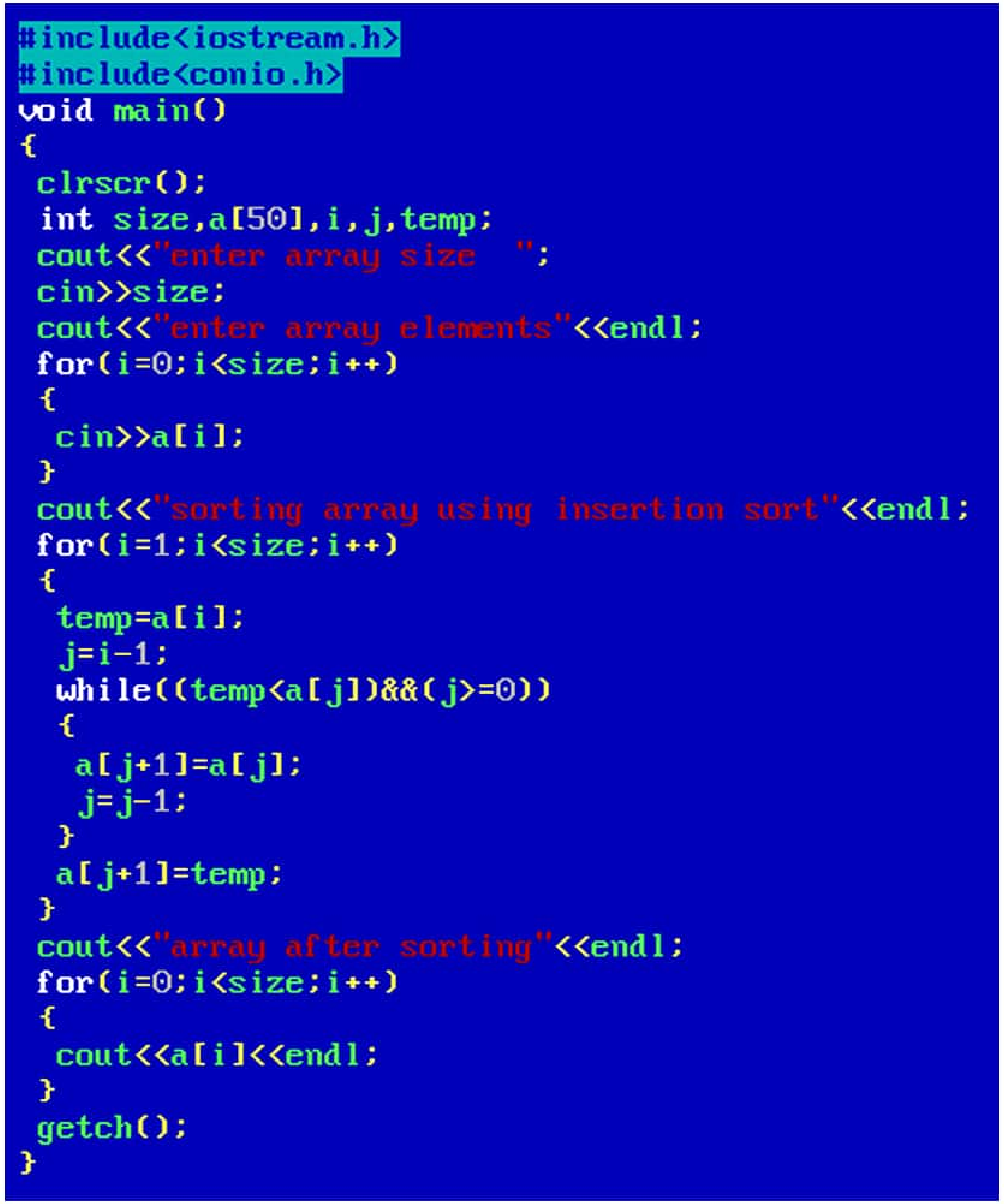


```
 front=front->next;
}
void queue :: display()
{
 for(node *temp=front;temp!=0;temp=temp->next)
 cout<<temp->data<<endl;
}
void main()
{
 int item,c;
 clrscr();
 queue q;
 do
 {
  cout<<"1.insert 2.remove 3.display 4.exit"<<endl;
  cin>>c;
  switch(c)
  {
   case 1:
   cout<<"enter item"<<endl;
   cin>>item;
   q.insert(item);
   break;
   case 2:
   q.remove();
   break;
   case 3:
   q.display();
   break;
   default:
   exit(0);
  }
  getch();
 }
 while(1);
}
```

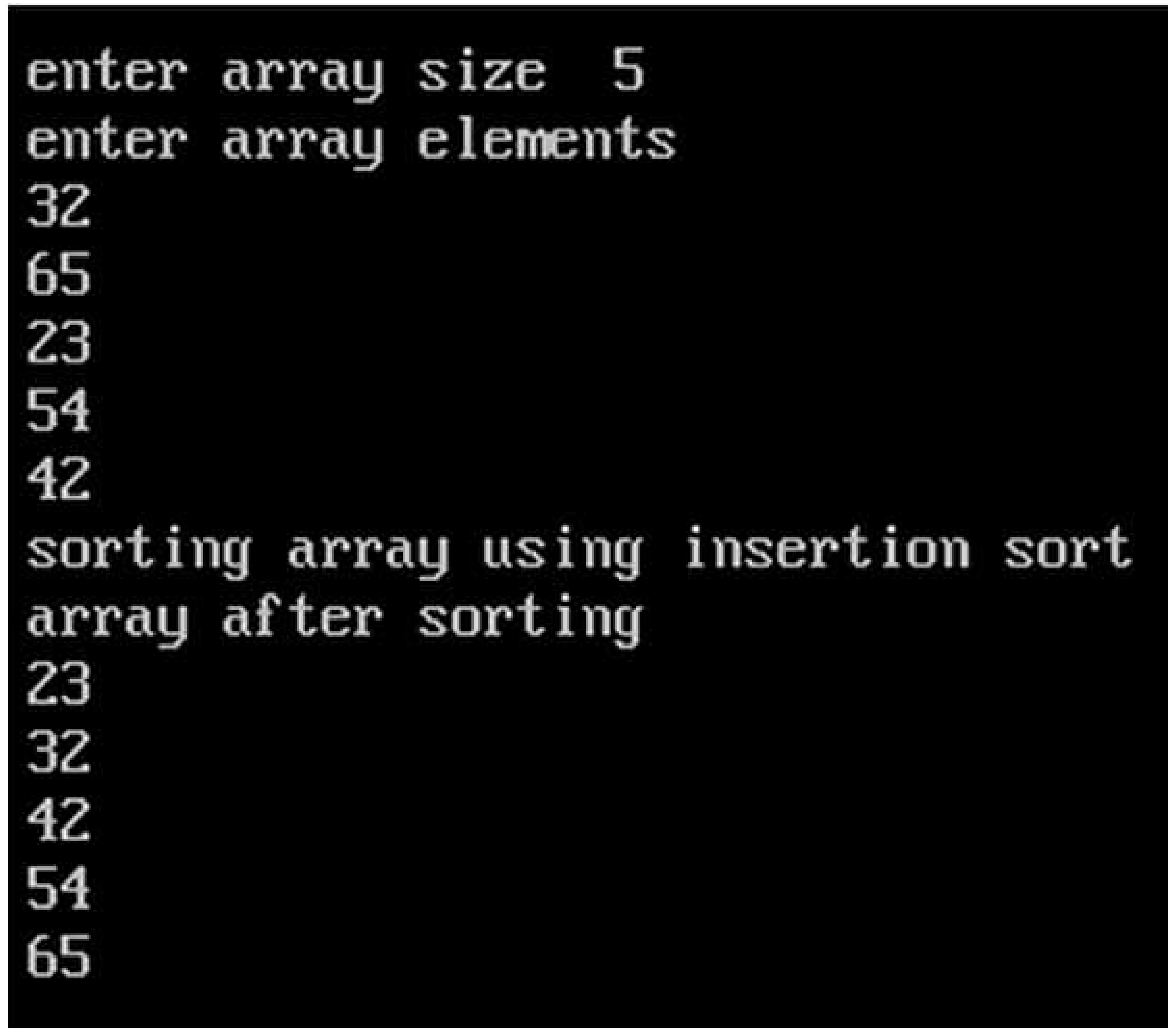


```
1.insert 2.remove 3.display 4.exit
1
enter item
10
1.insert 2.remove 3.display 4.exit
1
enter item
20
1.insert 2.remove 3.display 4.exit
3
10
20
1.insert 2.remove 3.display 4.exit
2
deleted 10
1.insert 2.remove 3.display 4.exit
2
deleted 20
1.insert 2.remove 3.display 4.exit
2
queue is empty
1.insert 2.remove 3.display 4.exit
4
```

insertion sort

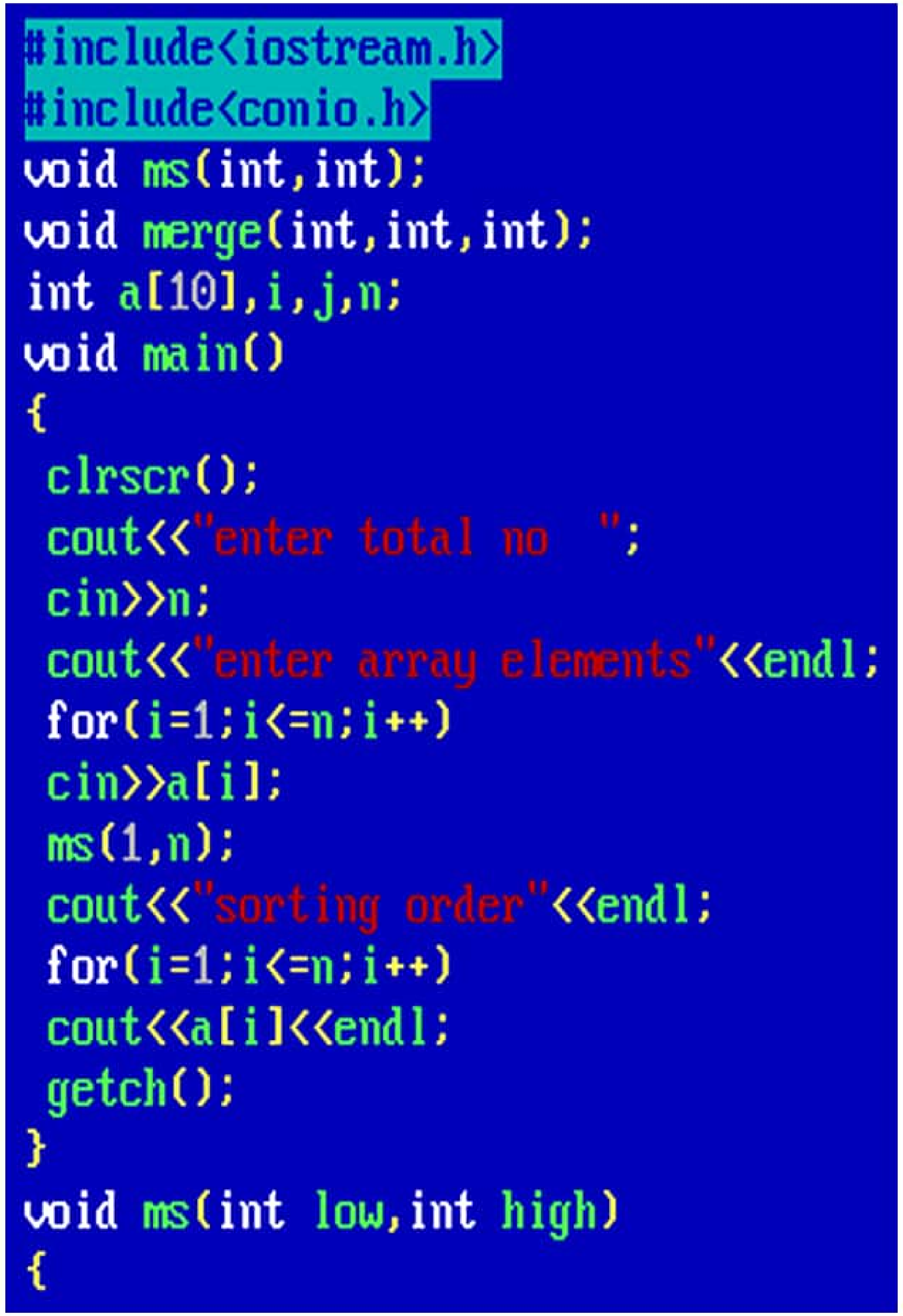


```
#include<iostream.h>
#include<conio.h>
void main()
{
 clrscr();
 int size,a[50],i,j,temp;
 cout<<"enter array size  ";
 cin>>size;
 cout<<"enter array elements"<<endl;
 for(i=0;i<size;i++)
 {
  cin>>a[i];
 }
 cout<<"sorting array using insertion sort"<<endl;
 for(i=1;i<size;i++)
 {
  temp=a[i];
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

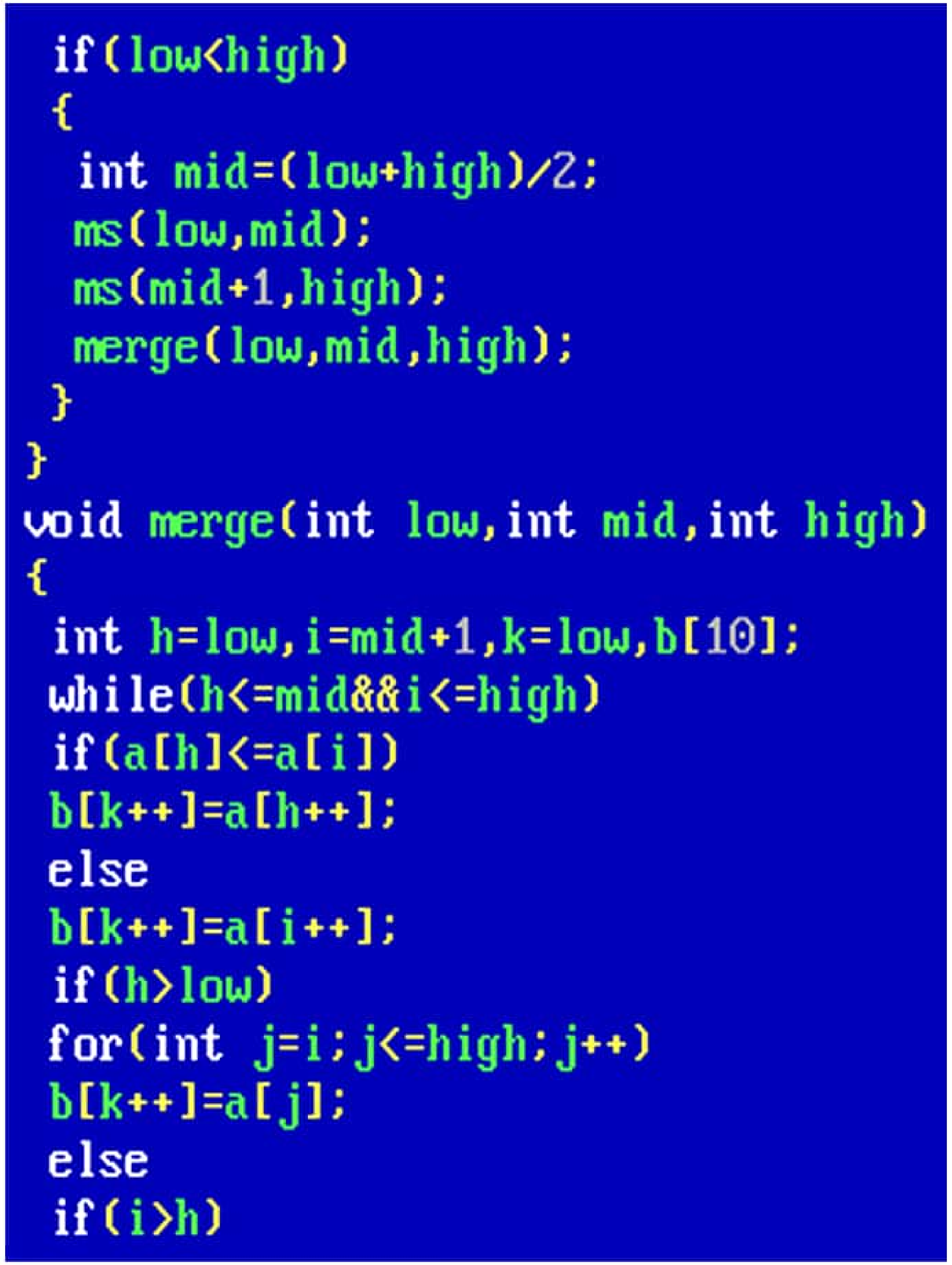


```
enter array size  5
enter array elements
32
65
23
54
42
sorting array using insertion sort
array after sorting
23
32
42
54
65
```

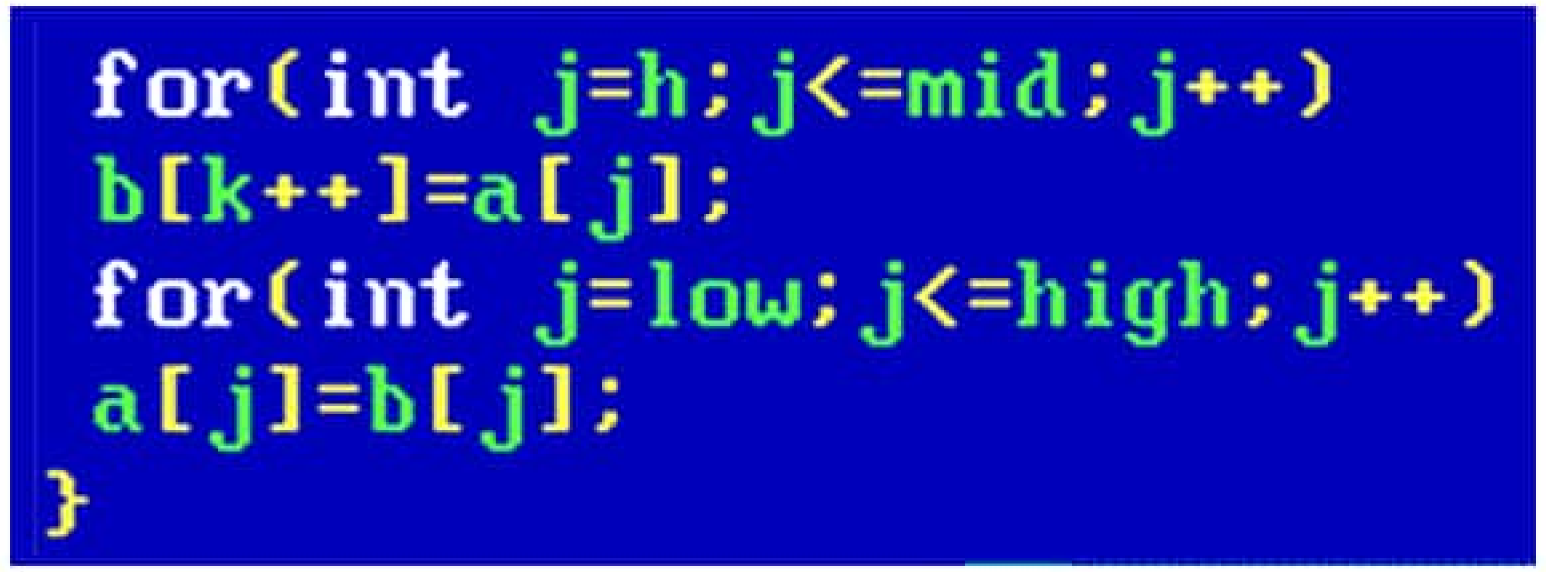
merge sort



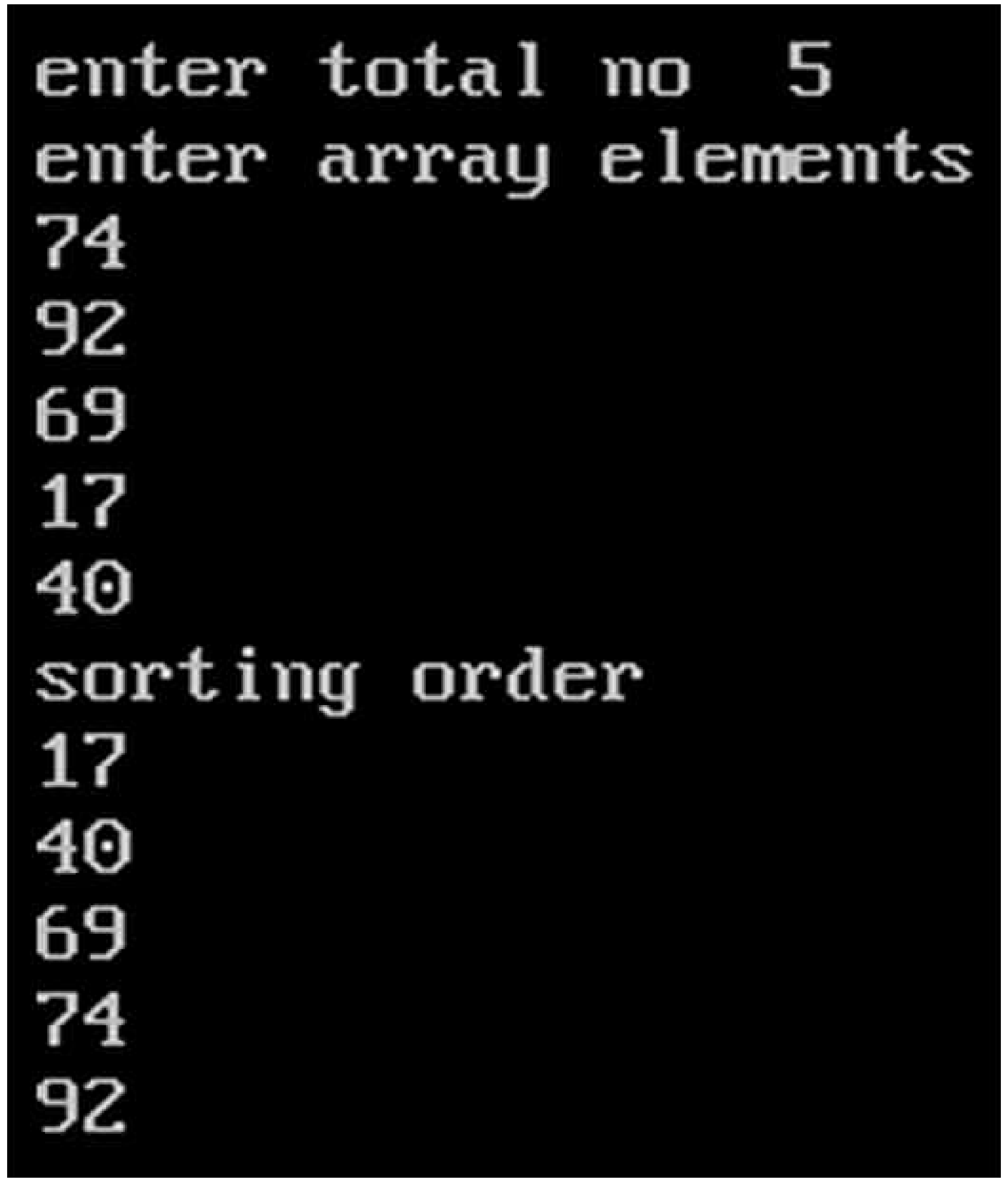
```
#include<iostream.h>
#include<conio.h>
void ms(int,int);
void merge(int,int,int);
int a[10],i,j,n;
void main()
{
 clrscr();
 cout<<"enter total no  ";
 cin>>n;
 cout<<"enter array elements"<<endl;
 for(i=1;i<=n;i++)
 cin>>a[i];
 ms(1,n);
 cout<<"sorting order"<<endl;
 for(i=1;i<=n;i++)
 cout<<a[i]<<endl;
 getch();
}
void ms(int low,int high)
{
```



```
  if(low<high)
  {
   int mid=(low+high)/2;
   ms(low,mid);
   ms(mid+1,high);
   merge(low,mid,high);
  }
}
void merge(int low,int mid,int high)
{
  int h=low,i=mid+1,k=low,b[10];
  while(h<=mid&&i<=high)
  if(a[h]<=a[i])
  b[k++]=a[h++];
  else
  b[k++]=a[i++];
  if(h>low)
  for(int j=i;j<=high;j++)
  b[k++]=a[j];
  else
  if(i>h)
```

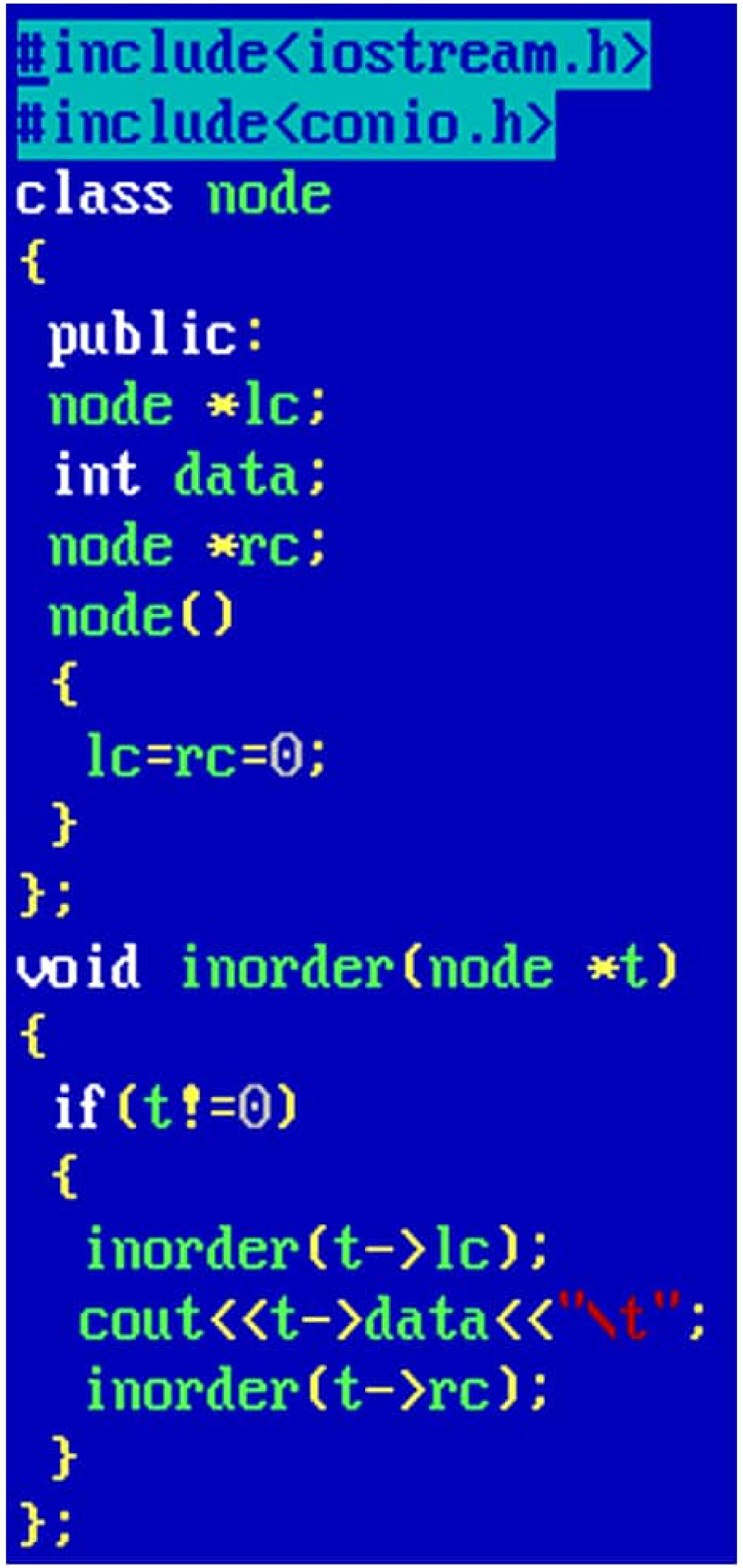


```
 for(int j=h;j<=mid;j++)
 b[k++]=a[j];
 for(int j=low;j<=high;j++)
 a[j]=b[j];
}
```

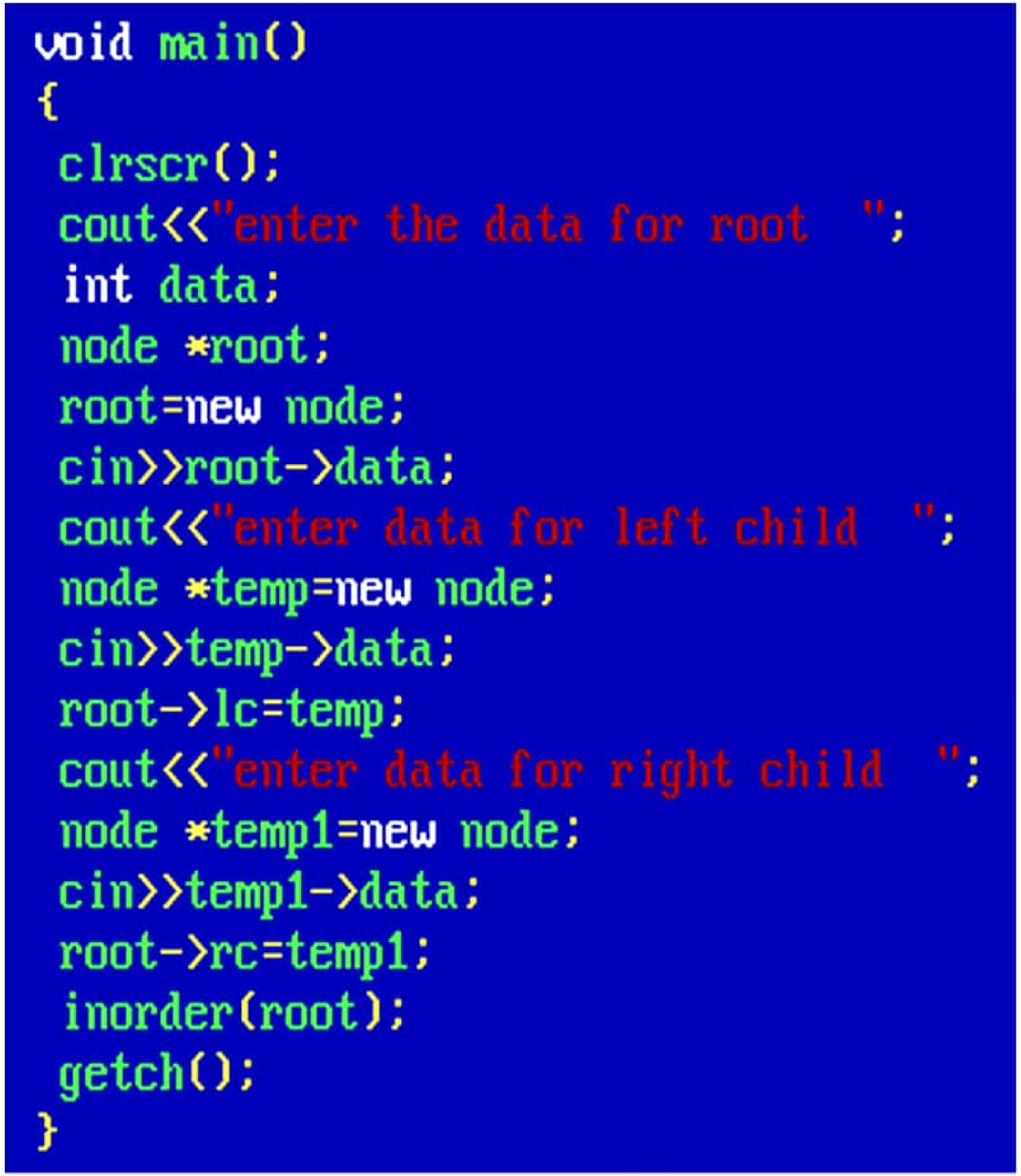


```
enter total no  5
enter array elements
74
92
69
17
40
sorting order
17
40
69
74
92
```

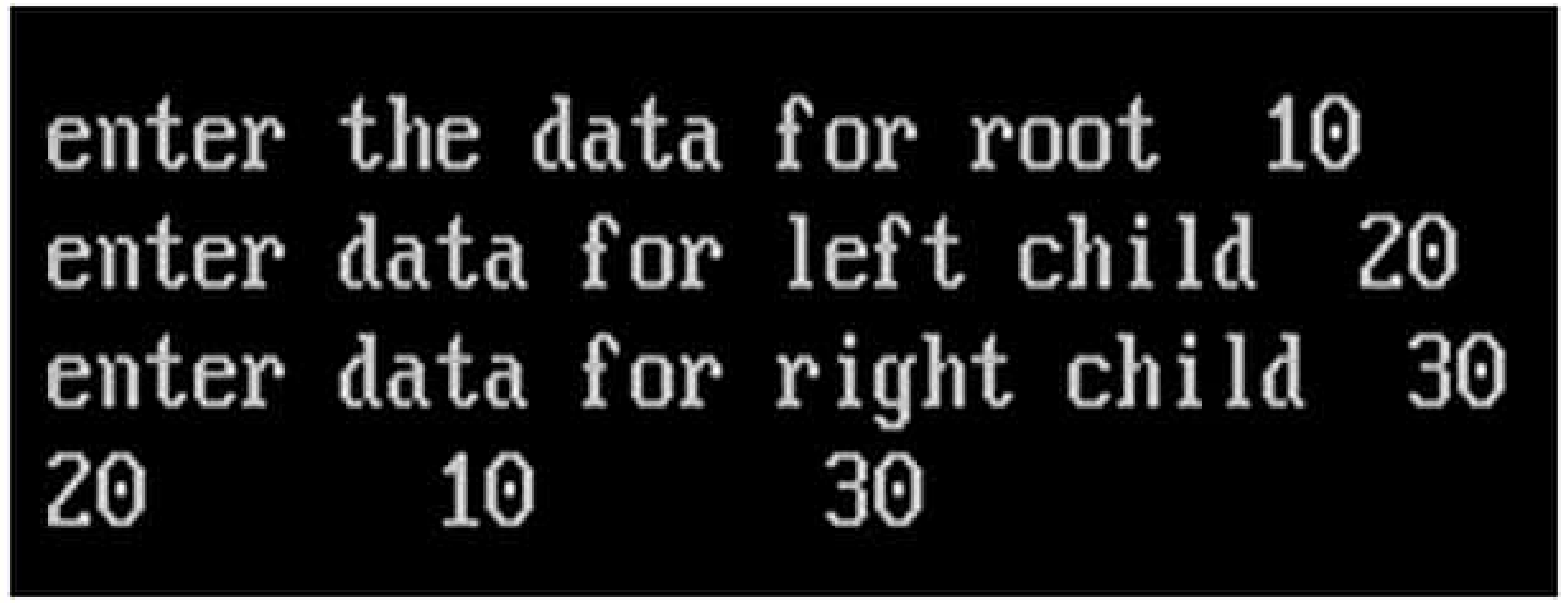
inorder



```
#include<iostream.h>
#include<conio.h>
class node
{
 public:
 node *lc;
 int data;
 node *rc;
 node()
 {
  lc=rc=0;
 }
};
void inorder(node *t)
{
 if(t!=0)
 {
  inorder(t->lc);
  cout<<t->data<<"\t";
  inorder(t->rc);
 }
};
```

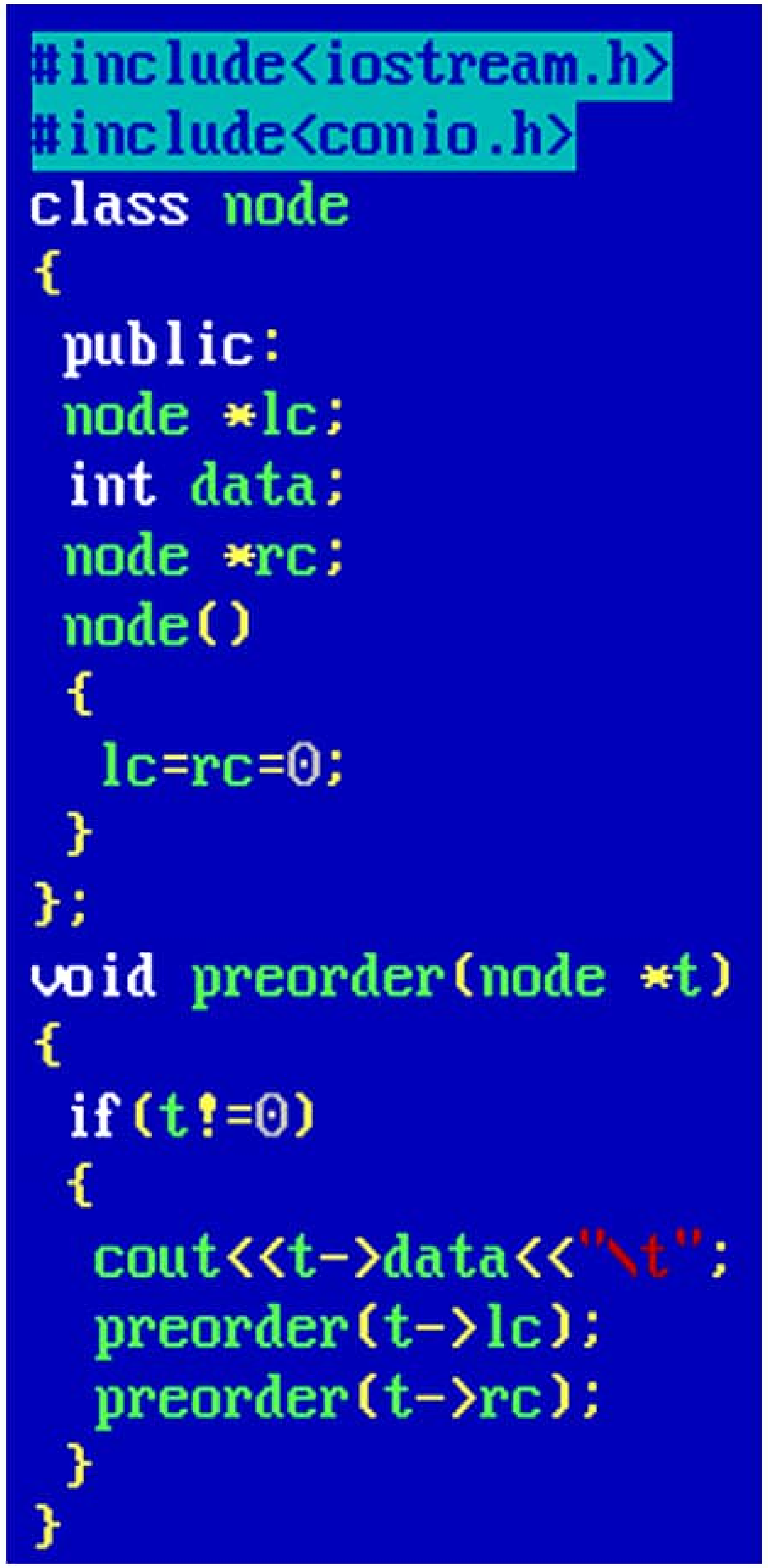


```
void main()
{
 clrscr();
 cout<<"enter the data for root  ";
 int data;
 node *root;
 root=new node;
 cin>>root->data;
 cout<<"enter data for left child  ";
 node *temp=new node;
 cin>>temp->data;
 root->lc=temp;
 cout<<"enter data for right child  ";
 node *temp1=new node;
 cin>>temp1->data;
 root->rc=temp1;
 inorder(root);
 getch();
}
```

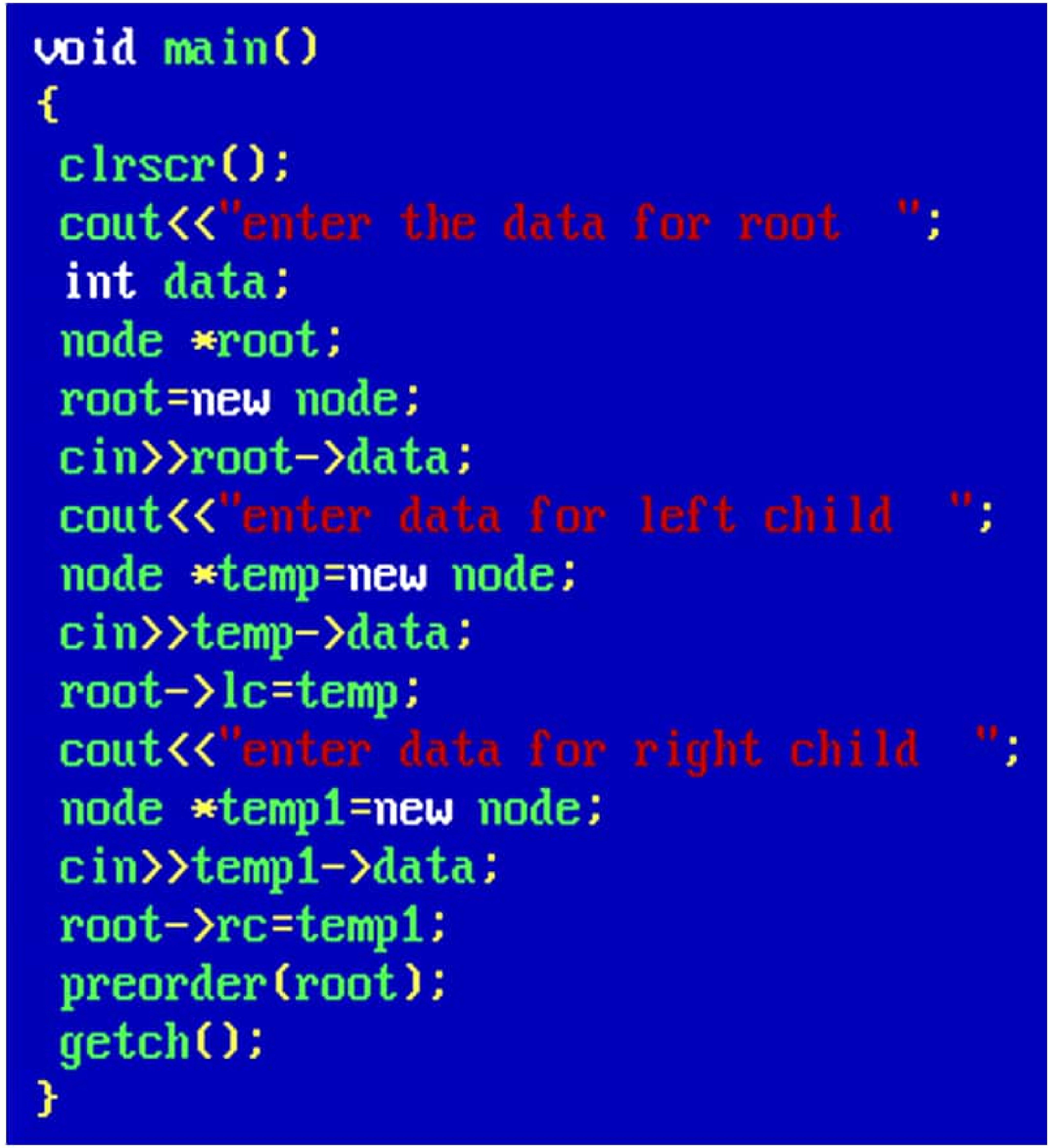


```
enter the data for root  10
enter data for left child  20
enter data for right child  30
20      10      30
```

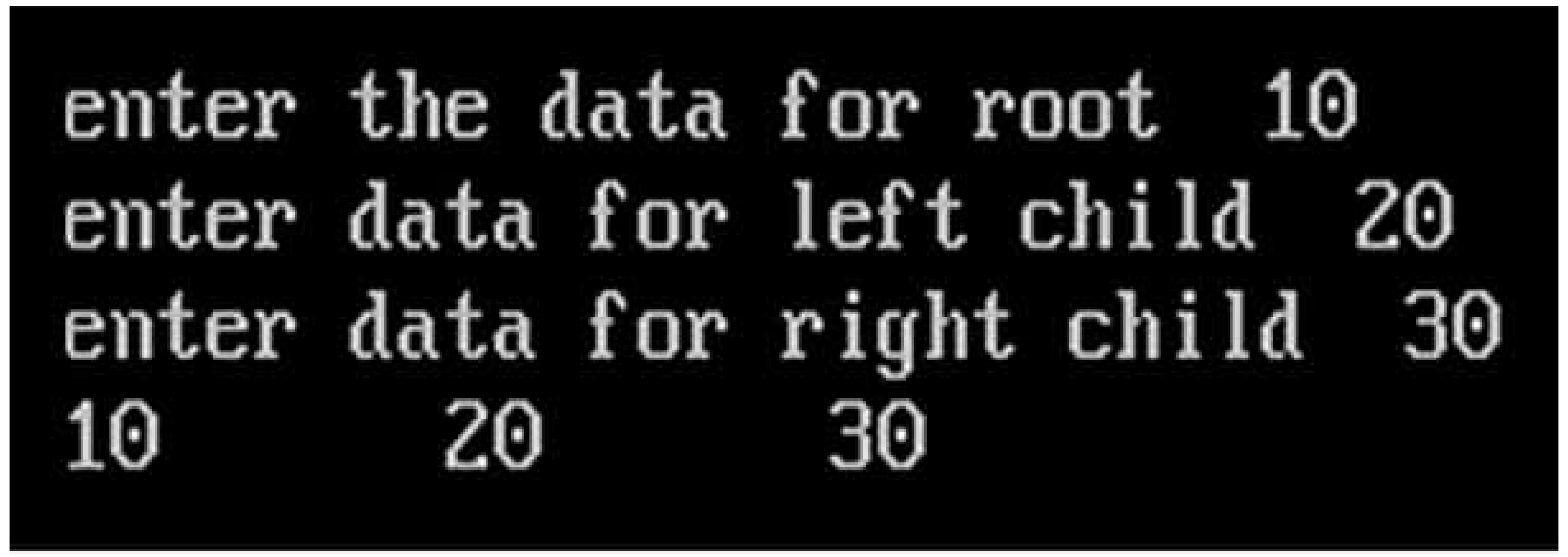
preorder



```
#include<iostream.h>
#include<conio.h>
class node
{
 public:
 node *lc;
 int data;
 node *rc;
 node()
 {
  lc=rc=0;
 }
};
void preorder(node *t)
{
 if(t!=0)
 {
  cout<<t->data<<"\t";
  preorder(t->lc);
  preorder(t->rc);
 }
}
```

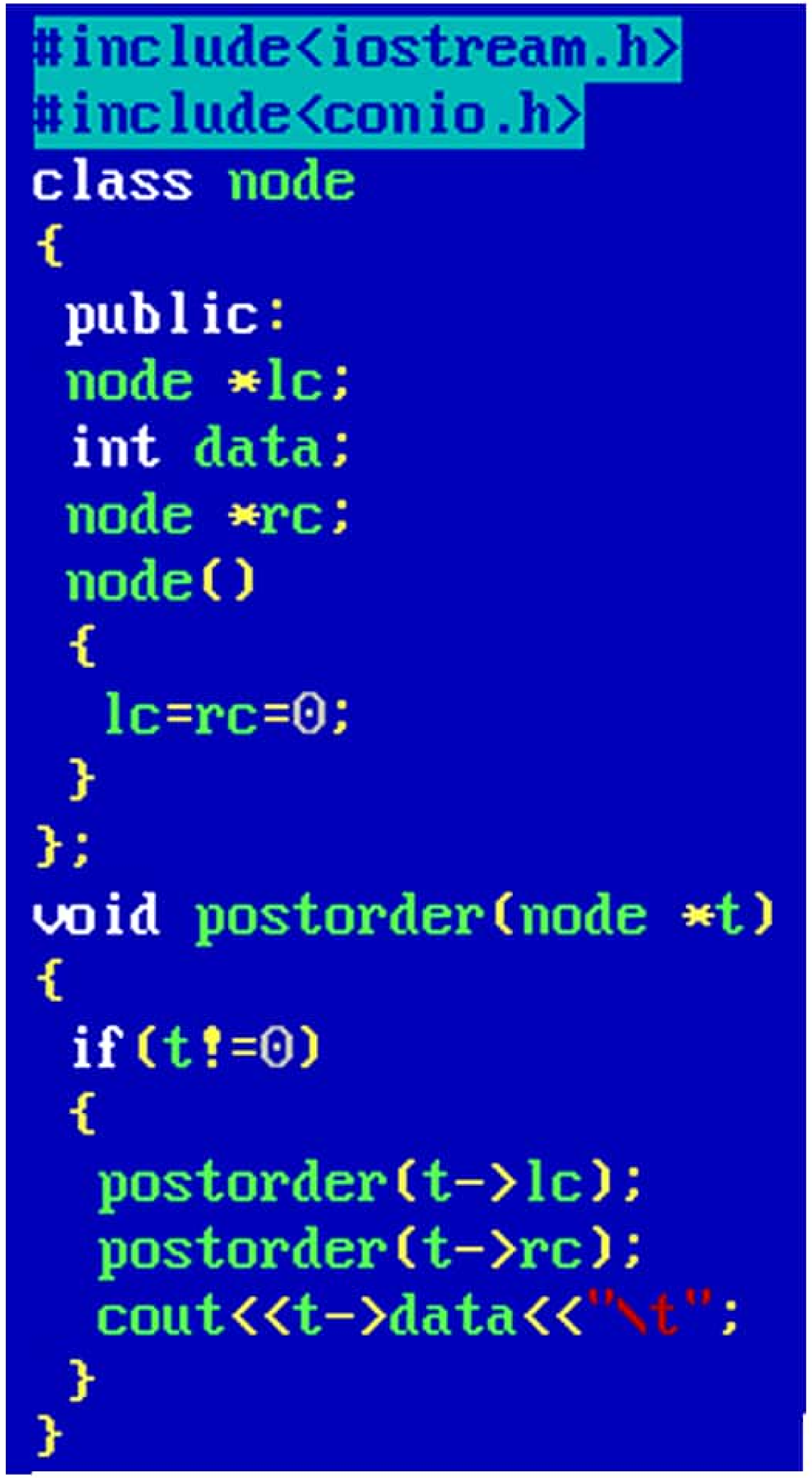


```
void main()
{
 clrscr();
 cout<<"enter the data for root  ";
 int data;
 node *root;
 root=new node;
 cin>>root->data;
 cout<<"enter data for left child  ";
 node *temp=new node;
 cin>>temp->data;
 root->lc=temp;
 cout<<"enter data for right child  ";
 node *temp1=new node;
 cin>>temp1->data;
 root->rc=temp1;
 preorder(root);
 getch();
}
```

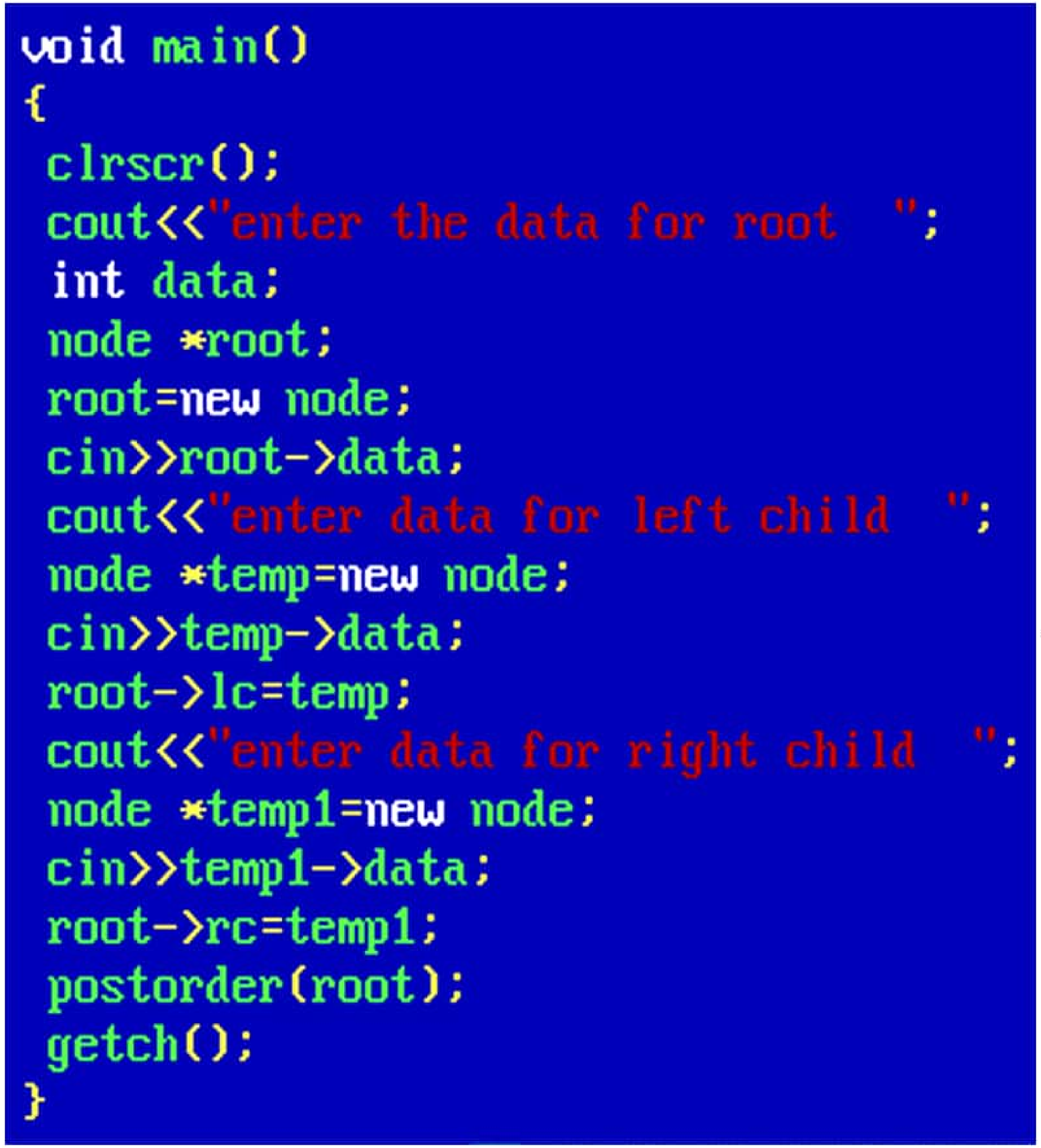


```
enter the data for root  10
enter data for left child  20
enter data for right child  30
10      20      30
```

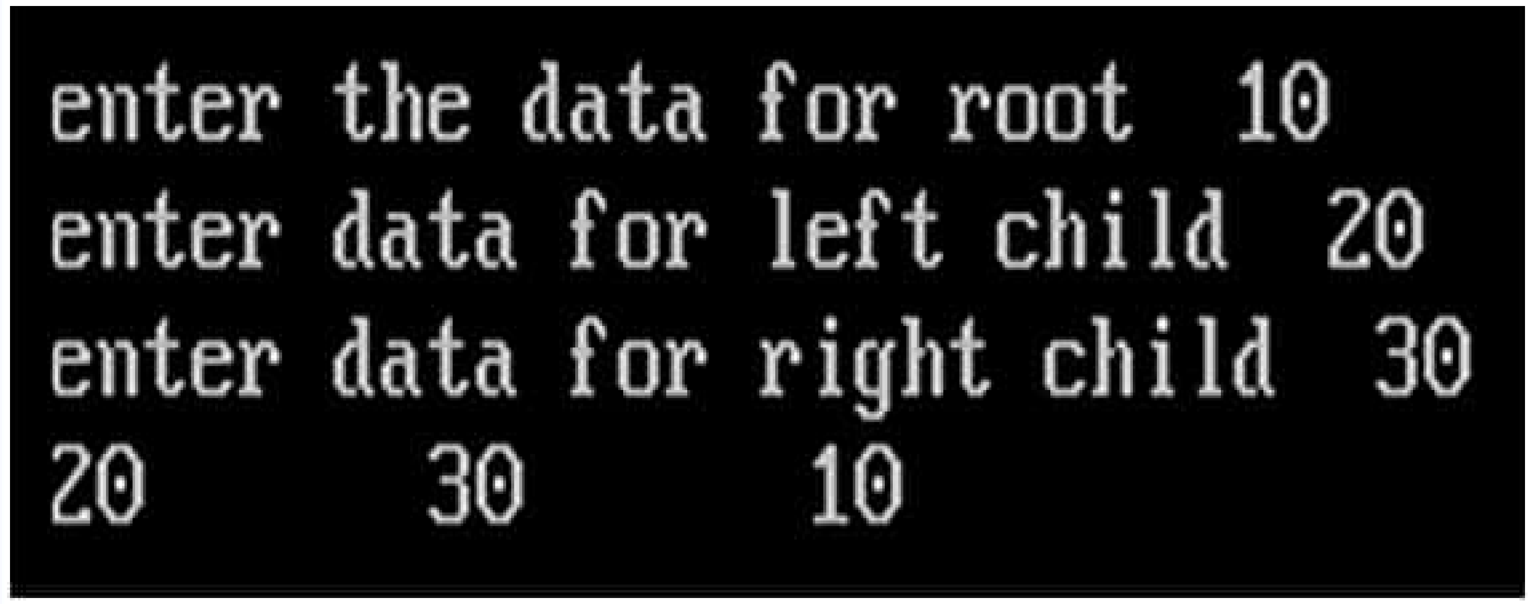
postorder



```
#include<iostream.h>
#include<conio.h>
class node
{
 public:
 node *lc;
 int data;
 node *rc;
 node()
 {
  lc=rc=0;
 }
};
void postorder(node *t)
{
 if(t!=0)
 {
  postorder(t->lc);
  postorder(t->rc);
  cout<<t->data<<"\t";
 }
}
```



```
void main()
{
 clrscr();
 cout<<"enter the data for root  ";
 int data;
 node *root;
 root=new node;
 cin>>root->data;
 cout<<"enter data for left child  ";
 node *temp=new node;
 cin>>temp->data;
 root->lc=temp;
 cout<<"enter data for right child  ";
 node *temp1=new node;
 cin>>temp1->data;
 root->rc=temp1;
 postorder(root);
 getch();
}
```



```
enter the data for root  10
enter data for left child  20
enter data for right child  30
20      30      10
```