

KRE SOCIETY'S



KARNATAK ARTS, SCIENCE AND COMMERCE COLLEGE, BIDAR

DEPARTMENT OF COMPUTER SCIENCE

PRACTICAL ASSIGNMENTS

CLASS: BSC VI SEM

PAPER: DATA STRUCTURE WITH C++

INDEX

SL.NO	NAME OF PROGRAMS	PAGE NO	REMARKS
1	PROGRAM TO INSERT AN ELEMENT INTO A LINEAR ARRAY		
2	PROGRAM TO DELETE AN ELEMENT FROM A LINEAR ARRAY		
3	PROGRAM TO DETERMINE WHETHER INPUT MATRIX IS A SPARSE MATRIX OR NOT		
4	PROGRAM TO IMPLEMENT BUBBLE SORT		
5	PROGRAM TO IMPLEMENT QUICK SORT		
6	PROGRAM IMPLEMENT INSERTION SORT		
7	PROGRAM IMPLEMENT RADIX SORT		
8	PROGRAM TO IMPLEMENT LINEAR SEARCH		
9	PROGRAM TO SEARCH THE ELEMENTS USING BINARY SEARCH		
10	PROGRAM TO CONVERT INFIX TO POSTFIX EXPRESION		
11	PROGRAM FOR EVALUTION OF POSTFIX EXPRESSION		
12	PROGRAM TO IMPLEMENT OPERATIONS ON STACK		
13	PROGRAM TO PERFORM OPERATIONS ON STACK USING DYNAMIC IMPLEMENTATION		
14	PROGRAM TO IMPLEMENT OPERATIONS ON QUEUE		
15	PROGRAM TO PERFORM ENQUE AND DEQUE OPERATION ON THE CIRCULAR QUEUE		
16	PROGRAM TO INSERT AN ELEMENT AT END OF A SINGLE LINKED LIST		
17	PROGRAM TO SEARCH AN ELEMENT IN SINGLE LINKED LIST		
18	PROGRAM TO DELET AN ELEMENT AT END POSITION IN DOUBLE LINKED LIST		
19	PROGRAM TO SORT THE GIVEN ELEMENTS IN DOUBLE LINKED LIST		
20	PROGRAM TO PRIFIX ORDER TRAVERSAL USING BINARY TREE		

```
/*1. PROGRAM TO INSERT AN ELEMENT INTO A LINEAR ARRAY*/
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[10],n,k,item,i;
```

```
cout<<"ENTER THE SIZE OF ARRAY \n";
```

```
cin>>n;
```

```
cout<<"ENTER THE ELEMENTS \n";
```

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

```
cin>>a[i];
```

```
cout<<"ENTER THE POSITION TO INSERT AN ELEMENT \n";
```

```
cin>>k;
```

```
cout<<"ENTER THE ELEMENT \n";
```

```
cin>>item;
```

```
i=n;
```

```
while(i>=k)
```

```
{
```

```
a[i+1]=a[i];
```

```
i--;
```

```
}
```

```
a[k]=item;
```

```
n=n+1;
```

```
cout<<"ARRAY AFTER INSERTION \n";
```

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

```
cout<<a[i]<<endl;
```

```
getch();
```

```
}
```

-----OUTPUT-----

ENTER THE SIZE OF ARRAY

5

ENTER THE ELEMENTS

1

2

3

4

5

ENTER THE POSITION TO INSERT AN ELEMENT

2

ENTER THE ELEMENT

10

ARRAY AFTER INSERTION

1

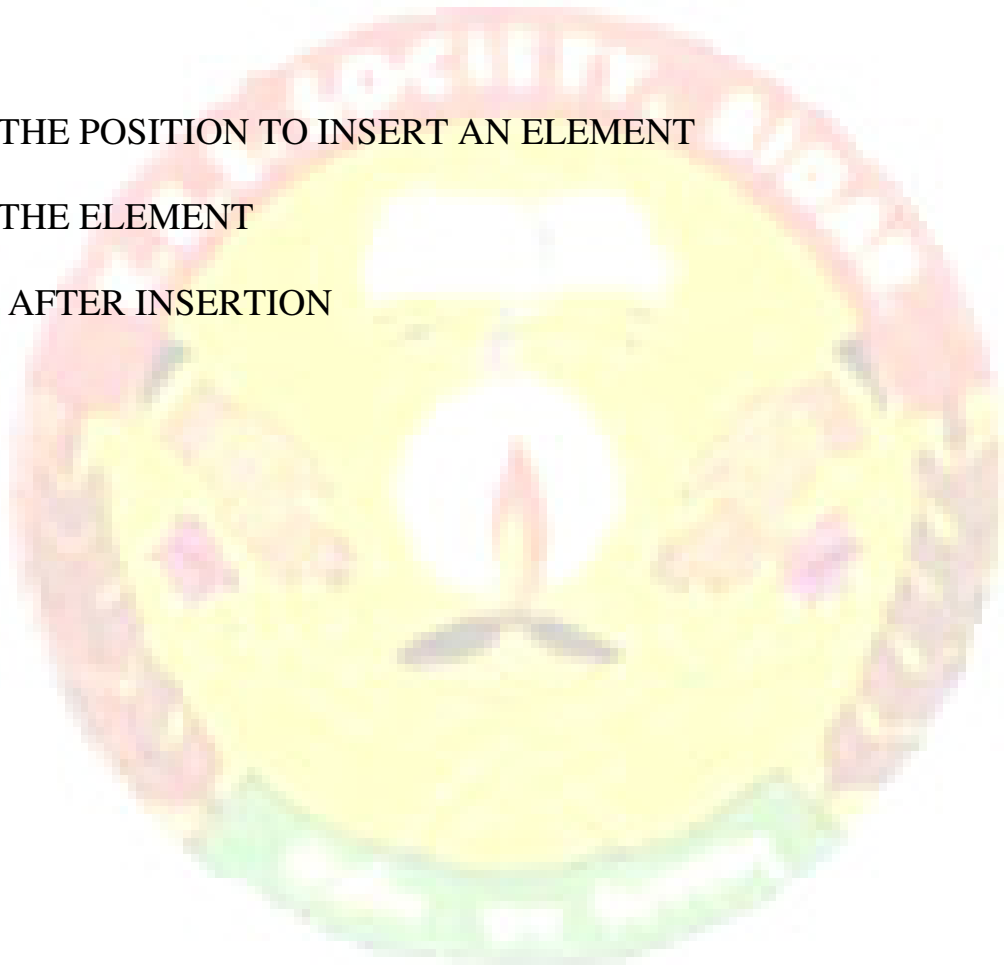
10

2

3

4

5



```
/*2. PROGRAM TO DELETE AN ELEMENT FROM A LINEAR ARRAY */
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[10],n,k,item,i;
```

```
cout<<"ENTER THE SIZE OF ARRAY \n";
```

```
cin>>n;
```

```
cout<<"ENTER THE ELEMENTS \n";
```

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

```
cin>>a[i];
```

```
cout<<"ENTER THE POSITION OF ELEMENT TO BE DELETED \n";
```

```
cin>>k;
```

```
item=a[k];
```

```
for(i=k;i<=n;i++)
```

```
{
```

```
a[i]=a[i+1];
```

```
}
```

```
n=n-1;
```

```
cout<<"THE DELETED ELEMENT= "<<item<<endl;
```

```
cout<<"ARRAY AFTER DELETION \n";
```

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

```
cout<<a[i]<<endl;
```

```
getch();
```

```
}
```

-----OUTPUT-----

ENTER THE SIZE OF ARRAY

5

ENTER THE ELEMENTS

1

2

3

4

5

ENTER THE POSITION OF ELEMENT TO BE DELETED

2

THE DELETED ELEMENT=2

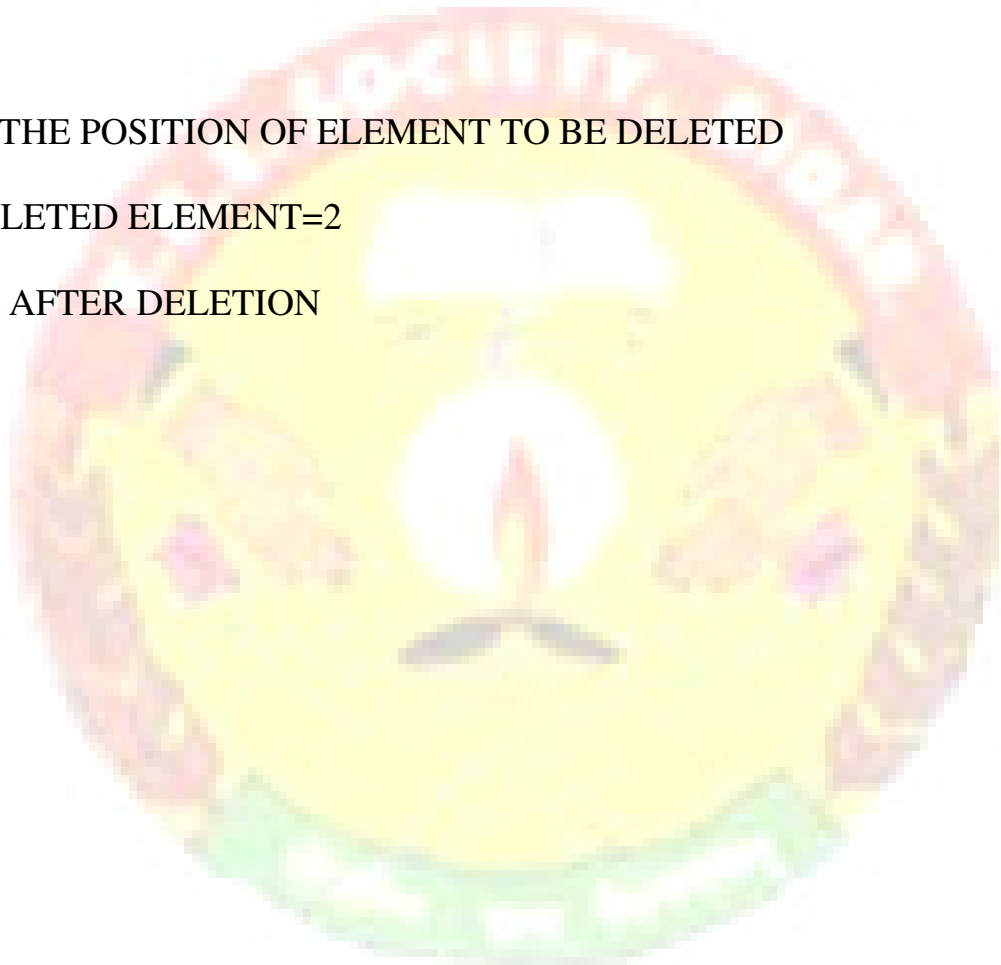
ARRAY AFTER DELETION

1

3

4

5



```
/* 3. PROGRAM TO DETERMINE WHETHER INPUT MATRIX IS A  
SPARSE MATRIX OR NOT */
```

```
#include<iostream.h>  
#include<conio.h>
```

```
void main ()
```

```
{
```

```
clrscr();
```

```
int i,j,m,n;
```

```
int counter=0;
```

```
static int m1[10][10];
```

```
cout<<" \n ENTER THE ORDER OF MATRIX \n";
```

```
cin>>m>>n;
```

```
cout<<" ENTER THE CO-EFFICIENTS OF THE MATRIX \n";
```

```
for (i=0;i<m;++i)
```

```
{
```

```
for (j=0;j<n;++j)
```

```
{
```

```
cin>>m1[i][j];
```

```
if (m1[i][j]==0)
```

```
++counter;
```

```
}
```

```
}
```

```
if (counter>((m*n)/2))
```

```
{
```

```
cout<<"\n THE GIVEN MATRIX IS A SPARSE MATRIX \n";
```

```
}
```

```
else
```

```
cout<<"\n THE GIVEN MATRIX IS NOT A SPARSE MATRIX \n ";
```

```
cout<<"\n NUMBER OF ZEROS=" <<counter;
```

```
getch();
```

```
}
```

-----OUTPUT-----

ENTER THE ORDER OF MATRIX

2

2

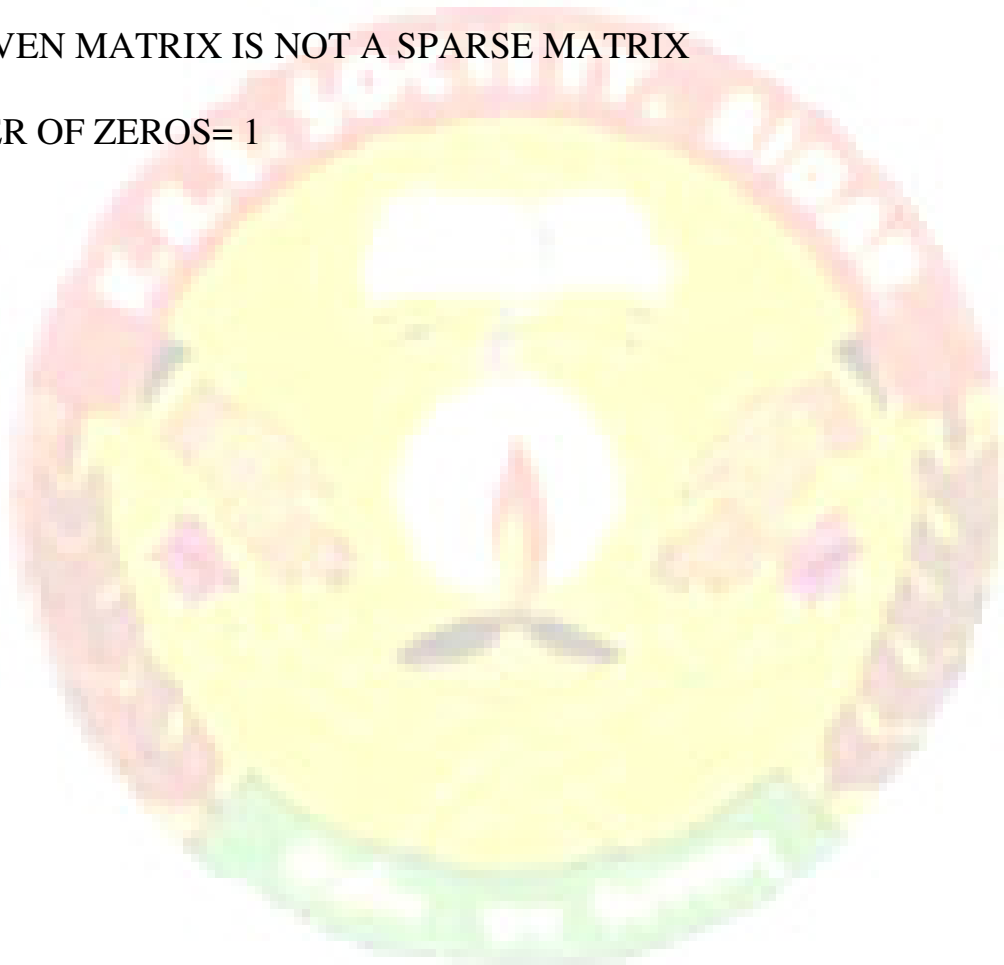
ENTER THE CO-EFFICIENTS OF THE MATRIX

1 2

0 3

THE GIVEN MATRIX IS NOT A SPARSE MATRIX

NUMBER OF ZEROS= 1



```
/* 4. PROGRAM TO IMPLEMENT BUBBLE SORT */
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[10],n,i,j,temp;
```

```
cout<<"ENTER THE SIZE OF ARRAY \n";
```

```
cin>>n;
```

```
cout<<"ENTER THE ELEMENTS \n";
```

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

```
cin>>a[i];
```

```
for(j=0;j<n;j++)
```

```
{
```

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

```
{
```

```
if(a[i]>a[i+1])
```

```
{
```

```
temp=a[i];
```

```
a[i]=a[i+1];
```

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

```
}
```

```
}
```

```
}
```

```
cout<<"ARRAY AFTER BUBBLE SORT \n";
```

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

```
cout<<a[i]<<endl;
```

```
getch();
```

```
}
```


-----OUTPUT-----

ENTER THE SIZE OF ARRAY

6

ENTER THE ELEMENTS

1

6

5

2

4

3

ARRAY AFTER BUBBLE SORT

1

2

3

4

5

6



```
/* 5.PROGRAM TO IMPLEMENT QUICK SORT */
```

```
#include<iostream.h>
#include<conio.h>
#include<dos.h>
#define max 100
void quicksort(int[],int,int);

void main()
{
int a[max],i,n;
clrscr();
cout<<"-----OUTPUT-----\n";
cout<<"Enter the limit\n";
cin>>n;
cout<<"Enter the elements\n";

for(i=0;i<=n-1;i++)
{
cin>>a[i];
}

quicksort(a,0,n-1);
cout<<"The sorted elements\n";
for(i=0;i<=n-1;i++)
{
cout<<a[i];
cout<<"\n";
}
}
```

```
void quicksort(int a[],int lb,int ub)
{
int i,j,temp,key,flag=1;
if(lb<ub)
{
i=lb;
j=ub+1;
key=a[lb];

while(flag)
{
i++;
while(a[i]<key&& i<ub)
i++;
j--;
while(a[j]>key)
j--;
if (i<j)
{
temp = a[i];
a[i] = a[j];
a[j] = temp;
}
else
flag = 0;
}
a[lb] = a[j];
a[j] = key;
quicksort(a,lb,j-1);
quicksort(a,j+1,ub);
}
}
```



-----OUTPUT-----

Enter the limit

5

Enter the elements

4

56

65

55

2

The sorted elements

2

4

55

56

65



```
/* 6.PROGRAM IMPLEMENT INSERTION SORT */
```

```
#include<iostream.h>
#include<conio.h>
void main()
{
int i,j,k,n,a[10],temp,ptr;
clrscr();
cout<<"-----OUTPUT-----\n";
cout<<"Enter the limit\n";
cin>>n;
cout<<"Enter the element of array\n";
for(i=1;i<=n;i++)
{
cin>>a[i];
}
a[0] = 0;
for(k=2;k<=n;k++)
{
temp = a[k];
ptr = k-1;
while(temp<a[ptr])
{
a[ptr+1] = a[ptr];
ptr = ptr-1;
a[ptr+1] = temp;
}
}
cout<<"The sorted elements are\n";
for(j=1;j<=n;j++)
{
cout<<a[j]<<endl;

}
getch();
}
```

-----OUTPUT-----

Enter the limit

5

Enter the element of array

40

10

20

60

50

The sorted elements are

10

20

40

50

60



```
/* 7.PROGRAM IMPLEMENT RADIX SORT */
```

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#define SIZE 10
#define DIGIT 3

int main()

{
int a[SIZE],n,i,j,p,temp;
clrscr();

cout<<"Enter the no. of terms:";
cin>>n;

cout<<"Enter the terms:\n";
for(i=0;i<n;i++)
cin>>a[i];

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

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

if((a[j] % (int) pow(10,p))>(a[j+1]%(int) pow(10,p)))
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
cout <<"The sorted array is\n";
for (i=0 ; i < n; i++)
cout << endl << a[i];
getch();
return(0);
}
```

-----OUTPUT-----

Enter the no. of terms:5

Enter the terms:

1

5

4

2

3

The sorted array is

1

2

3

4

5




```
/*8. PROGRAM TO IMPLEMENT LINEAR SEARCH */
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int a[10],n,i,loc,item;
```

```
cout<<"ENTER THE SIZE OF ARRAY \n";
```

```
cin>>n;
```

```
cout<<"ENTER THE ELEMENTS \n";
```

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

```
cin>>a[i];
```

```
cout<<"ENTER THE ELEMENT TO BE SEARCHED \n";
```

```
cin>>item;
```

```
a[n+1]=item;
```

```
loc=1;
```

```
while(a[loc]!=item)
```

```
loc=loc+1;
```

```
if(loc==n+1)
```

```
cout<<"ELEMENT IS NOT FOUND \n" ;
```

```
else
```

```
cout<<"SEARCH IS SUCCESSFULL \n";
```

```
cout<<" ITEM IS FOUND AT POSITION="<<loc;
```

```
getch();
```

```
}
```

-----OUTPUT-----

ENTER THE SIZE OF ARRAY

5

ENTER THE ELEMENTS

1

3

5

6

8

ENTER THE ELEMENT TO BE SEARCHED

5

SEARCH IS SUCCESSFULL

ITEM IS FOUND AT POSITION=3

ENTER THE SIZE OF ARRAY

5

ENTER THE ELEMENTS

10

20

30

40

50

ENTER THE ELEMENT TO BE SEARCHED

15

ELEMENT IS NOT FOUND

```
/* 9.PROGRAM TO SERACH THE ELEMENTS USING BINARY SEARCH */
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
class bin
```

```
{
```

```
    int a[10],i,n,k,e,b,m;
```

```
public:
```

```
    void search();
```

```
    void in();
```

```
    void out();
```

```
};
```

```
void bin::in()
```

```
{
```

```
    cout<<"-----OUTPUT-----\n";
```

```
    cout<<"Enter the array size\n";
```

```
    cin>>n;
```

```
    cout<<"Enter the elements in ascending order\n";
```

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

```
{
```

```
    cin>>a[i];
```

```
}
```

```
    cout<<"Enter the elements to be searched\n";
```

```
    cin>>k;
```

```
}
```

```
void bin::search()
```

```
{
```

```
b = 1;
```

```
e = n;
```

```
m = (b+e)/2;
```

```
while((b<=e)&&(a[m]!=k))
```

```
{
```

```
    if(k<a[m])
```

```
        e = m-1;
```

```
    else
```

```
        b = m+1;
```

```
        m = (b+e)/2;
```

```
}
```

```
}  
void bin::out()  
{  
if(k==a[m])  
{  
cout<<"Search is successful\n";  
}  
else  
{  
cout<<"Search is unsuccessful\n";  
}  
}  
void main()  
{  
bin b;  
clrscr();  
b.in();  
b.search();  
b.out();  
}
```

-----OUTPUT-----

```
Enter the array size  
5  
Enter the elements in ascending order  
3  
4  
5  
6  
7  
Enter the elements to be searched  
4  
Search is successful
```

```
/* 10. PROGRAM TO CONVERT INFIX TO POSTFIX EXPRESION*/
```

```
#include<iostream.h>
#include<ctype .h><
#include<string.h>
#include<conio.h>
#define max 50
class exp
{
    char infix[max],suffix[max],s[max];
    int top;
    public:
        exp(char *str);
        void push(char ch);
        char pop();
        void convert();
        int priority(char ch);
        void display();
};
exp::exp(char *str)
{
    strcpy(infix,str);
    top=0;
}
void exp::push(char ch)
{
    s[++top]=ch;
}
char exp::pop()
{
    return(s[top--]);
}

int exp::priority(char ch)
{
    if(ch=='('||ch=='#')
        return 1;
    if(ch=='+'||ch=='-')
        return 2;
```

```

    if(ch=='*'||ch=='\0')
        return 3;
    if(ch=='$'||ch=='^')
        return 4;
    return 0;
}
void exp::convert()
{
    int i,j=0;
    push('#');
    for(i=0;infix[i]!='\0';i++)
    {
        if(isalpa(infix[i]))
            suffix[j++]=infix[i];
        else if(infix[i]=='(')
            push(infix[i]);
        else if(infix[i]==')')
        {
            while(s[top]!='(')
                suffix[j++]=pop();
            pop();
        }
        else
        {
            while(priority(s[top])>=priority(infix[i]))
                suffix[j++]=pop();
            push(infix[i]);
        }
    }
    while(s[top]!='#')
    {
        suffix[j++]=pop();
        suffix[j]='\0';
    }
}
void exp::display()
{
    cout<<suffix;
}

```

```
}  
  
void main()  
{  
    char infix[50];  
    clrscr();  
    cout<<"-----OUTPUT-----\n";  
    cout<<"Enter infix expression";  
    cin>>infix;  
    exp p=infix;  
    p.convert();  
    cout<<"The postfix expression is: ";  
    p.display();  
}
```

-----OUTPUT-----

```
Enter infix expression  
a+b  
The postfix expression is: ab+
```

```
/* 11.PROGRAM FOR EVALUTION OF POSTFIX EXPRESSION */
```

```
#include<iostream.h>
#include<conio.h>
class post
{
    float stack[10],a,b;
    char item;
    public:void tree();
};
void post::tree()
{
    int top=0;
    cout<<"Program to evaluate a given postfix expression\n";
    cout<<"Enter the given Postfix in a single line\n";
    cout<<"&digit should be digit from 0to9\n";
    item=getch();
    while(item!='\n')
    {
        switch(item)
        {
            case'+':b=stack[top];
                a=stack[top-1];
                top--;
                stack[top]=float(a+b);
                break;
            case'-':b=stack[top];
                a=stack[top-1];
                top--;
                stack[top]=float(a-b);
                break;
            case'*':b=stack[top];
                a=stack[top-1];
                top--;
                stack[top]=float(a*b);
                break;
            case'/':b=stack[top];
                a=stack[top-1];
                top--;
```



```
        stack[top]=float(a/b);
        break;
default:top=top+1;
        stack[top]=item-'0';
    }
    cin>>item;
    }
    cout<<"The value of the given postfix exp is:"<<stack[top];
}
void main()
{
post p;
clrscr();
p.tree();
getch();
}
```

-----OUTPUT-----

Program to evaluate a given postfix expression
Enter the given Postfix in a single line
&digit should be digit from 0to9
(68+)
The value of the given postfix exp is: 14

```
/* 12.PROGRAM TO IMPLEMENT OPERATIONS ON STACK */
```

```
#include<iostream.h>  
#include<conio.h>  
#define size 5  
int top=0,ele,stack[size];
```

```
push()  
{  
if (top==size)  
{  
cout<<"\n STACK IS UNDERFLOW \n";  
}  
else  
{  
cout<<"\n ENTER THE ELEMENT \n";  
cin>>ele;  
top=top+1;  
stack[top]=ele;  
  
}  
return 0;  
}
```

```
pop()  
{  
if (top==0)  
{  
cout<<"\n STACK UNDERFLOW !!\n";  
}  
else  
{  
cout<<"\n POPPED ELEMENT IS : "<<stack[top];  
top=top-1;  
}  
return 0;  
}
```

```
display()
```

```

{
if (top==0)
{
cout<<"\n STACK UNDERFLOW!! \n";
}
else
{
cout<<"\n STACK ELEMENTS ARE \n";
for(int i=top;i>0;i--)
cout<<stack[i]<<endl;
}
return 0;
}

```

```

void main()
{
int ch;
clrscr();
do
{
cout<<"\n\n 1: PUSH\n 2: POP\n 3: STATUS\n 4: EXIT\n";
cout<<"\n ENTER YOUR CHOICE : ";
cin>>ch;
switch(ch)
{
case 1 :push();
break;
case 2 :pop();
break;
case 3 :display();
break;
case 4:break;

}
}while(ch!=4);
getch();
}

```

-----OUTPUT-----

1.PUSH

2.POP

3.STATUS

4.EXIT

ENTER YOUR CHOICE :1

ENTER THE ELEMENT

1

1.PUSH

2.POP

3.STATUS

4.EXIT

ENTER YOUR CHOICE :1

ENTER THE ELEMENT

2

1.PUSH

2.POP

3.STATUS

4.EXIT

ENTER YOUR CHOICE :1

ENTER THE ELEMENT

3

1.PUSH

2.POP

3.STATUS

4.EXIT

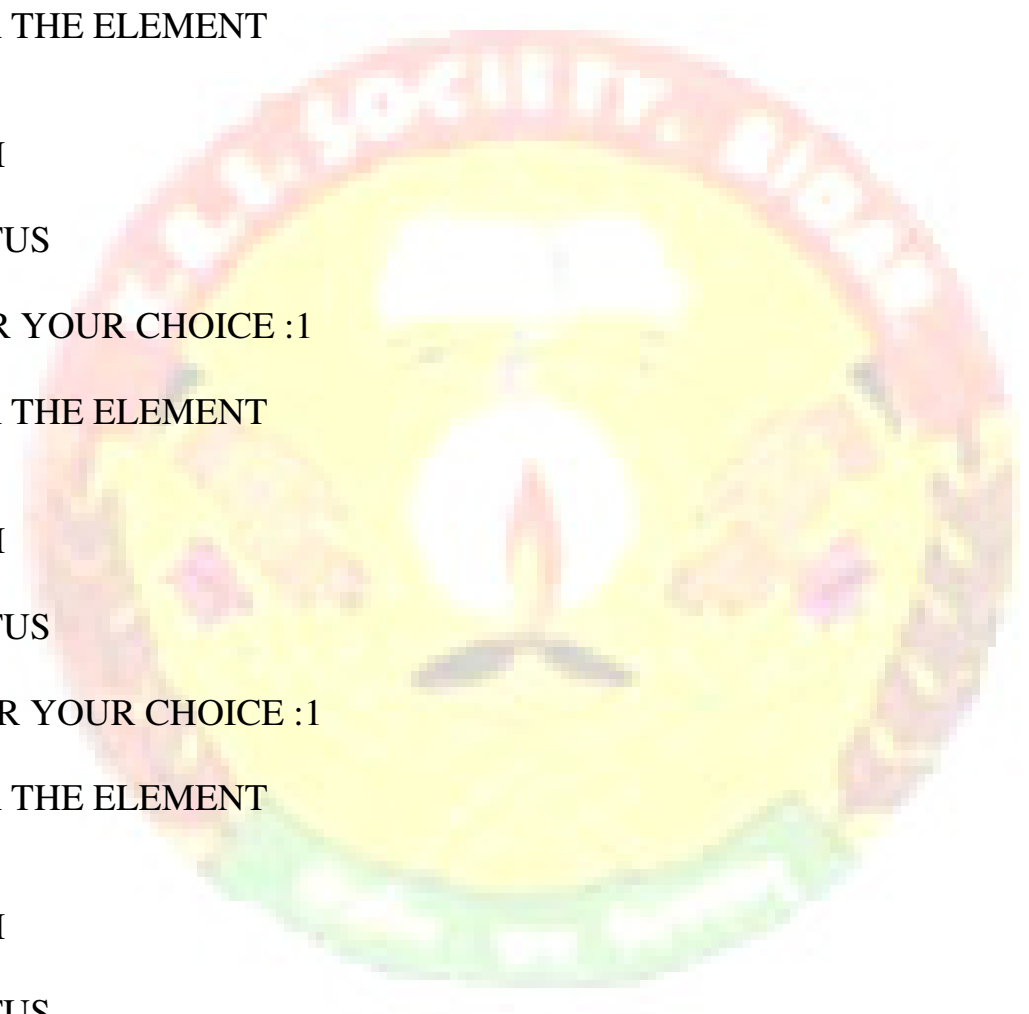
ENTER YOUR CHOICE :3

STACK ELEMENTS ARE

3

2

1



- 1.PUSH
- 2.POP
- 3.STATUS
- 4.EXIT

ENTER YOUR CHOICE :2
POPPED ELEMENT IS :3

- 1.PUSH
- 2.POP
- 3.STATUS
- 4.EXIT

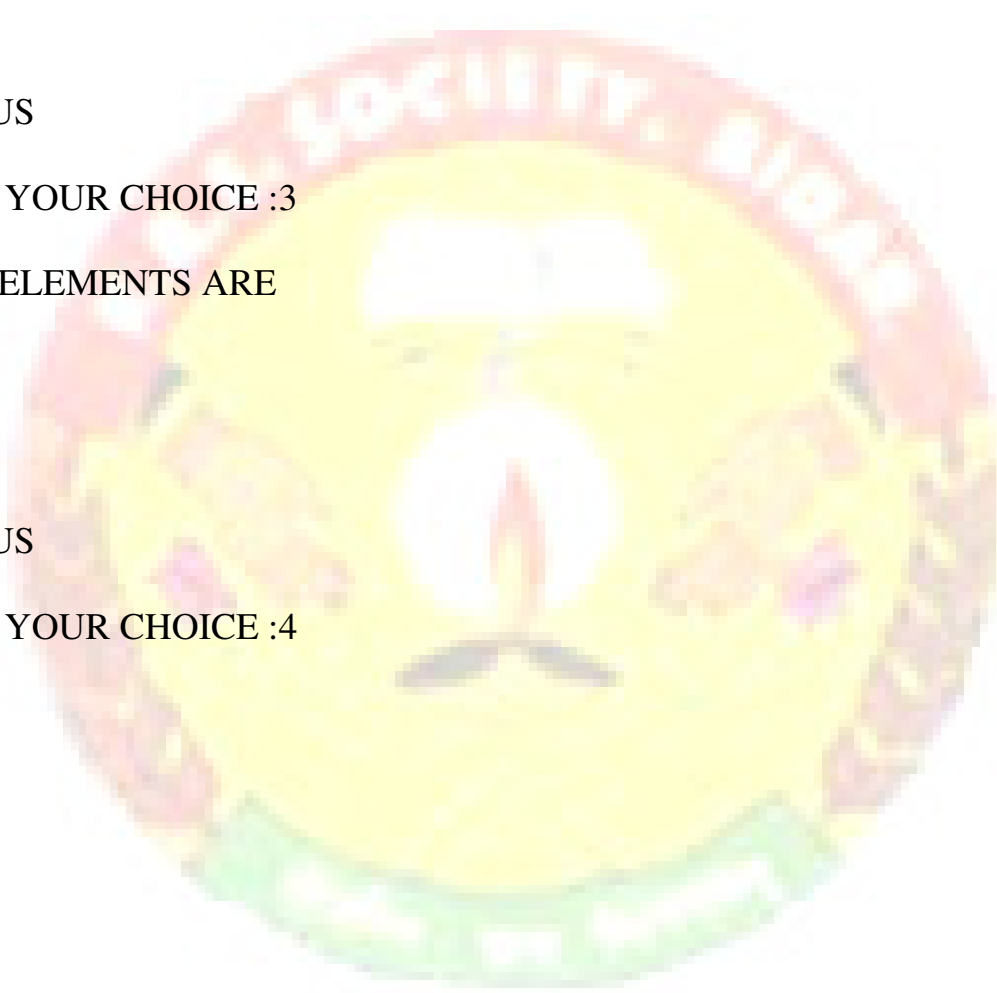
ENTER YOUR CHOICE :3

STACK ELEMENTS ARE

2
1

- 1.PUSH
- 2.POP
- 3.STATUS
- 4.EXIT

ENTER YOUR CHOICE :4



/*13. PROGRAM TO PERFORM OPERATIONS ON STACK USING
DYNAMIC IMPLEMENTATION*/

```
#include<iostream.h>
#include<conio.h>

struct node
{
    int data;
    node *link;
};
class stack
{
    int item,ch;
    node *head,*ptr,*pnew,*top;
public:
    void push();
    void pop();
    void show();

    stack()
    {
        head=new node;
        head->data=NULL;
        head->link=NULL;
    }
};
void stack::push()
{
    cout<<"\nEnter the item to be pushed in the stack \n ";
    cin>>item;
    if(head->link==NULL)
    {
        pnew=new node;
        head->link=pnew;
        pnew->data=item;
        pnew->link=NULL;
        top=pnew;
    }
}
```

```

else
{
    pnew=new node;
    head->link=pnew;
    pnew->data=item;
    pnew->link=top;
    top=pnew;
}
}
void stack::pop()
{
    top=head->link;
    if(head==NULL)
    {
        cout<<"\nStack is empty";
        goto a;
    }
    else
    {
        ptr=top->link;
        item=top->data;
        head->link=ptr;
        cout<<"\nThe popped element is: "<<top->data;
        top=ptr;
    }
a:
}
void stack::show()
{
    ptr=head->link;
    int nodecount=0;
    if(ptr==NULL)
    {
        cout<<"\nStack is empty\n";
        goto b;
    }
    else
    {
        while(ptr!=NULL)

```

```

        {
            nodecount++;
            cout<<ptr->data<<"\n";
            ptr=ptr->link;
        }
        cout<<"\nThe stack contains "<<nodecount<<" items";
        cout<<"\ Item at stack top contains  "<<top->data;
    }
b:
}

void main()
{
    int ch;
    char ans;
    stack ob;
    clrscr();
    do
    {
        cout<<"\n1.PUSH\n2.POP\n3.STATUS\n4.EXIT\n Enter your choice \n";
        cin>>ch;
        switch(ch)
        {
            case 1:ob.push();
                cout<<"\n The stack after pushing the element \n";
                ob.show();
                break;
            case 2:ob.pop();
                cout<<"\n The stack after popping the element \n";
                ob.show();
                break;
            case 3:cout<<"\nThe status of the stack is \n";
                ob.show();
            case 4:break;
        }
    }while(ch!=4);
    getch();
}

```


-----OUTPUT-----

- 1.PUSH
- 2.POP
- 3.STATUS
- 4.EXIT

Enter your choice

1

Enter the item to be pushed in the stack

1

The stack after pushing the element

1

The stack contains 1 items Item at stack top contains 1

1.PUSH

2.POP

3.STATUS

4.EXIT

Enter your choice

1

Enter the item to be pushed in the stack

2

The stack after pushing the element

2

The stack contains 2 items Item at stack top contains 2

1.PUSH

2.POP

3.STATUS

4.EXIT

Enter your choice

1

Enter the item to be pushed in the stack

3

The stack after pushing the element

3

2

1

The stack contains 3 items Item at stack top contains 3

- 1.PUSH
- 2.POP
- 3.STATUS
- 4.EXIT

Enter your choice

2

The popped element is: 3

The stack after popping the element

2

1

The stack contains 2 items Item at stack top contains 2

1.PUSH

2.POP

3.STATUS

4.EXIT

Enter your choice

3

The status of the stack is

2

1

The stack contains 2 items Item at stack top contains 2

1.PUSH

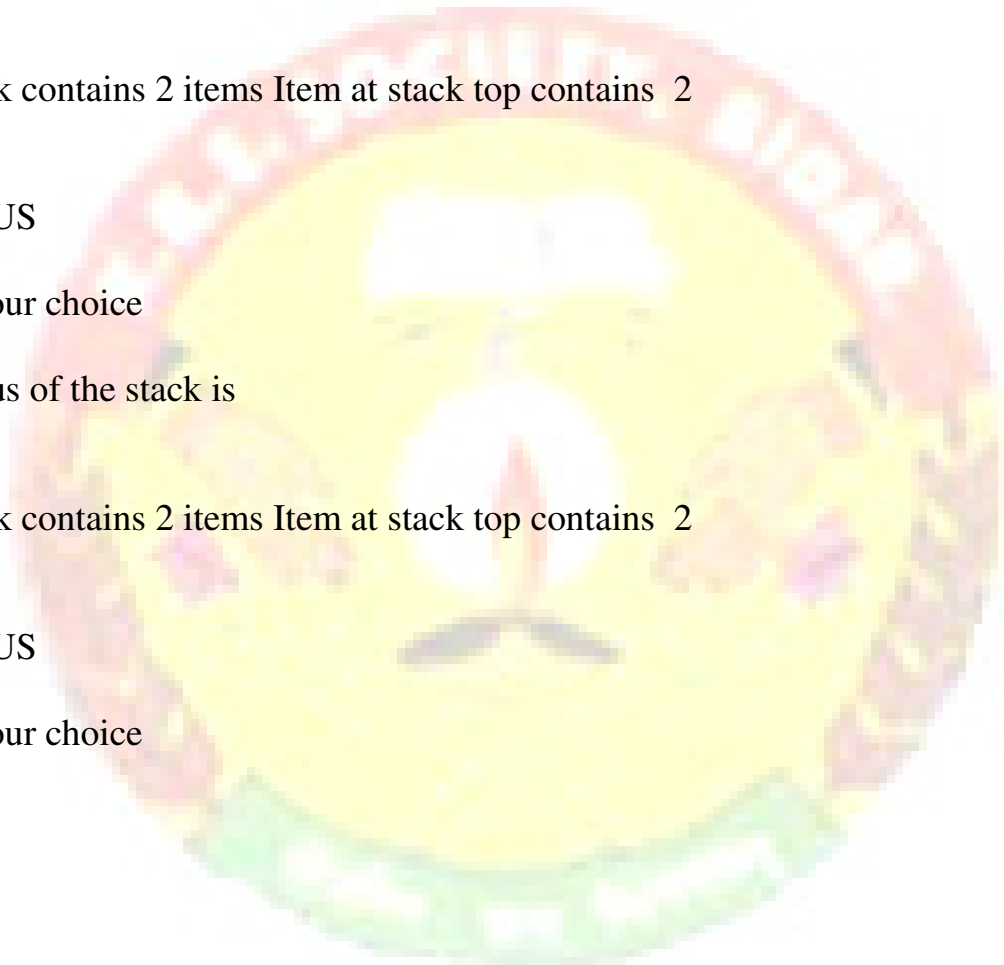
2.POP

3.STATUS

4.EXIT

Enter your choice

4



```
/* 14. PROGRAM TO IMPLEMENT OPERATIONS ON QUEUE */
```

```
#include<iostream.h>
#include<conio.h>
#define max 5
template<class dt>
class queue
{
    private:dt a[10],ele;
            int front,rear;
    public:queue();
            void add();
            void del();
            void disp();
};
template<class dt>
void queue<dt>::queue()
{
    front=rear=-1;
}

template<class dt>
void queue<dt>::add()
{
    if(rear==max-1)
    {
        cout<<" OVERFLOW OCCURS\n";
        return;
    }
    cout<<" ENTER THE ELEMENET \n";
    float ele;
    cin>>ele;
    rear=rear+1;
    a[rear]=ele;
    if(front==-1)
        front=0;
    return;
}
```

```

template<class dt>
void queue<dt>::del()
{
if(front==-1)
{
cout<<" UNDER FLOW OCCURS\n";
return;
}
cout<<" ELEMENT " <<a[front]<<" IS DELETED\n";
if(front==rear)
front=rear=-1;
else
front+=1;
}

```

```

template<class dt>
void queue<dt>::disp()
{
if(front==-1)
{
cout<<" EMPTY LIST \n";
return;
}
cout<<"\n ELEMENTS ARE:\n\n";
for(int i=front;i<=rear;i++)
cout<<a[i]<<" ";
}

```

```

void main()
{
queue<int> q;
queue<double> d;
int choice,ch,menu();
clrscr();
cout<<"\n 1:INTEGER ARRAY";
cout<<"\n 2:DOUBLE ARRAY\n";
cout<<"\n ENTER U R CHOICE \n";
cin>>ch;
switch(ch)

```

```

{
case 1:cout<<" INTEGER NUMBER PROCESSING \n";
    do
    {
    choice=menu();
    switch(choice)
    {
    case 1:q.add();
        q.disp();break;
    case 2:q.del();
        q.disp();break;
    case 3:q.disp();break;
    case 4:break;
    }
    }while(choice!=4);
    break;

case 2:cout<<" DOUBLE NUMBER PROCESSING\n";
    do
    {
    choice=menu();
    switch(choice)
    {
    case 1:d.add();
        d.disp();break;
    case 2:d.del();
        d.disp();break;
    case 3:d.disp();break;
    case 4:break;
    }
    }while(choice!=4);
    break;
default:cout<<" WRONG ENTRY\n";
    break;
}
getch();
}

int menu()

```

```
{
cout<<"\n\n 1:INSERT \n";
cout<<"\n 2:DELETE\n";
cout<<"\n 3:DISPLAY\n";
cout<<"\n 4:EXIT\n";
cout<<"\n ENTER YOUR CHOICE \n";
int ch1;
cin>>ch1;
return(ch1);
}
```

-----OUTPUT-----

```
1:INTEGER ARRAY
2:DOUBLE ARRAY
ENTER U R CHOICE
1
INTEGER NUMBER PROCESSING
1:INSERT
2:DELETE
3:DISPLAY
4:EXIT
ENTER YOUR CHOICE
1
ENTER THE ELEMENET
11
```

```
1:INSERT
2:DELETE
3:DISPLAY
4:EXIT
ENTER YOUR CHOICE
22
```

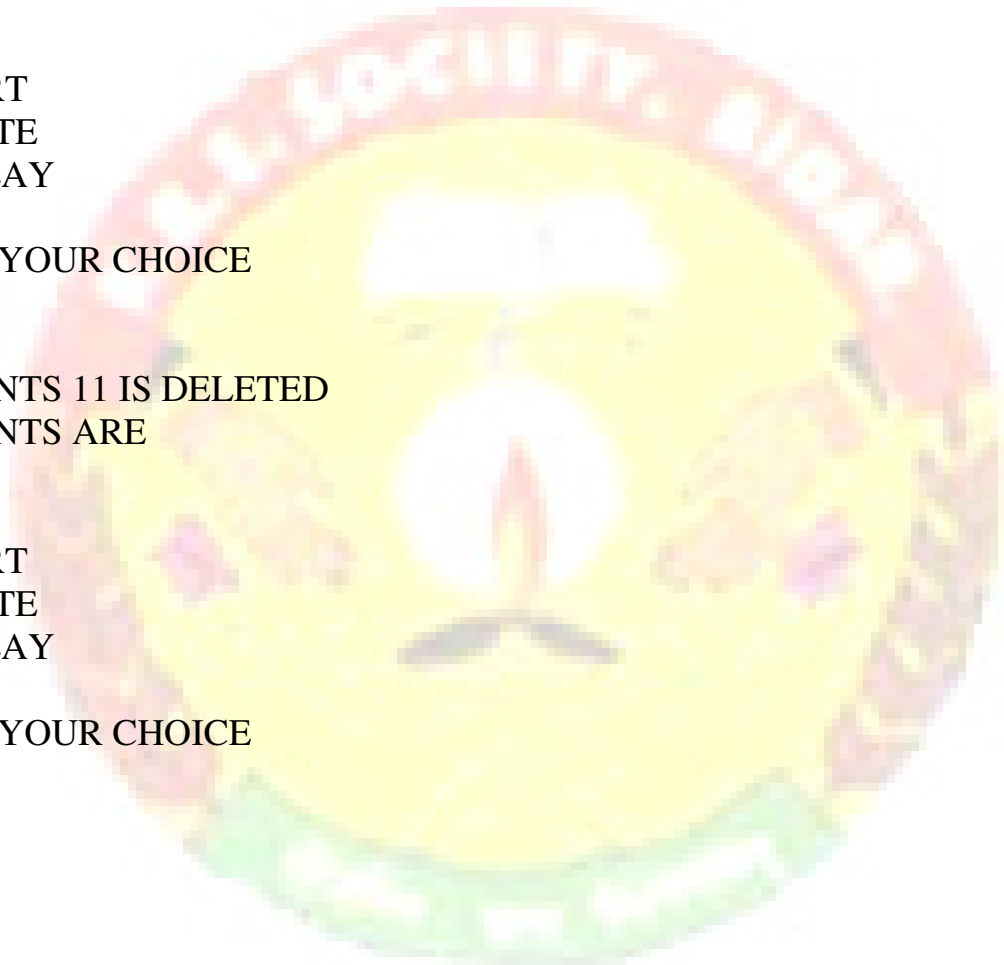
1:INSERT
2:DELETE
3:DISPLAY
4:EXIT
ENTER YOUR CHOICE
3

ELEMENTS ARE:
11 22

1:INSERT
2:DELETE
3:DISPLAY
4:EXIT
ENTER YOUR CHOICE
2

ELEMENTS 11 IS DELETED
ELEMENTS ARE
22

1:INSERT
2:DELETE
3:DISPLAY
4:EXIT
ENTER YOUR CHOICE
4



```
/* 15. PROGRAM TO PERFORM ENQUE AND DEQUE OPERATION ON  
THE CIRCULAR QUEUE */
```

```
#include<iostream.h>  
#include<stdlib.h>  
#include<conio.h>
```

```
struct node  
{  
    int data;  
    node *link;  
};  
class cirque  
{  
int item;  
node *head,*head1,*ptr,*ptr1;  
public:void create();  
    void display();  
    void insend();  
    void delf();  
};
```

```
void cirque::create()  
{  
int temp;  
head1=new node;  
ptr1=head1;  
cout<<"-----OUTPUT-----\n";  
cout<<"Enter the elements terminated by 0\n";  
cin>>temp;  
if(temp==0)  
{  
head1->link=head1;  
}  
while(temp!=0)  
{  
ptr=new node;  
ptr1->link=ptr;  
ptr->data=temp;
```

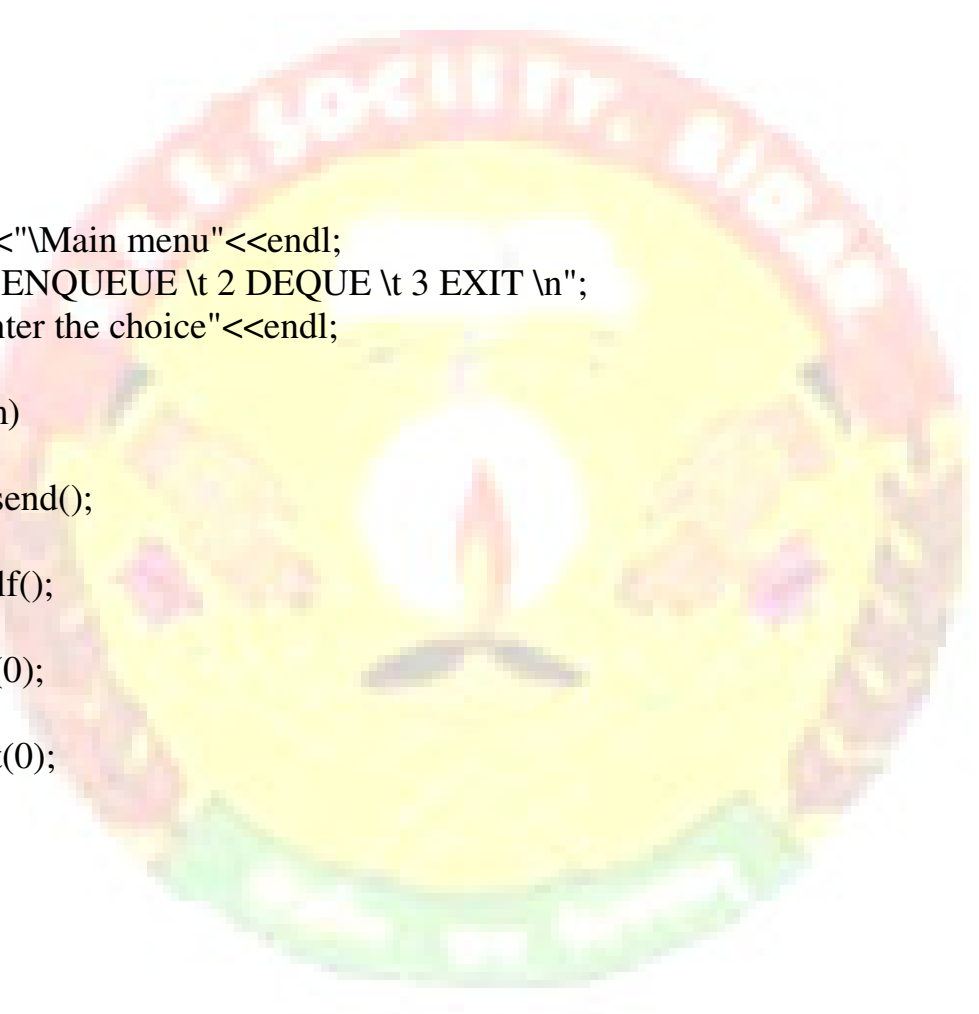


```

ptr1=ptr;
cin>>temp;
ptr->link=head1;
}
cout<<"\Created Queue in Linked list is"<<endl;
head=head1;
display();
}
void cirque::display()
{
if(head->link==head)
{
cout<<"\Queue is empty"<<endl;
exit(0);
}
ptr=head->link;
cout<<"FRONT->";
while(ptr!=head)
{
cout<<ptr->data<<"->";
ptr=ptr->link;
}
cout<<"REAR\n";
}
void cirque::insend()
{
cout<<"\Enter the key to br inserted"<<endl;
cin>>item;
ptr=head->link;
while(ptr->link!=head)
{
ptr=ptr->link;
}
node *newnode=new node;
newnode->data=item;
ptr->link=newnode;
newnode->link=head;
display();
}

```

```
void cirque::delf()
{
ptr=head->link;
ptr1=ptr->link;
head->link=ptr1;
display();
};
void main()
{
cirque c;
clrscr();
c.create();
int num;
start:cout<<"\Main menu"<<endl;
cout<<"\ 1 ENQUEUE \t 2 DEQUE \t 3 EXIT \n";
cout<<"\Enter the choice"<<endl;
cin>>num;
switch(num)
{
case 1:c.insend();
break;
case 2:c.delf();
break;
case 3:exit(0);
break;
default:exit(0);
}
goto start;
}
```



-----OUTPUT-----

Enter the elements terminated by 0

1
2
3
4
0

Created Queue in Linked list is

FRONT->1->2->3->4->REAR

Main menu

1 ENQUEUE 2 DEQUEUE

Enter the choice

1

Enter the key to be inserted

5

FRONT->1->2->3->4->5->REAR

Main menu

1 ENQUEUE 2 DEQUEUE

Enter the choice

2

FRONT->2->3->4->5->REAR

Main menu

1 ENQUEUE 2 DEQUEUE

Enter the choice

3

```
/*16. PROGRAM TO INSERT AN ELEMENT AT END OF A SINGLE  
LINKED LIST*/
```

```
#include<iostream.h>  
#include<conio.h>
```

```
struct node  
{  
    int data;  
    node *link;  
};
```

```
class sll  
{  
    node *ptr,*ptr1,*newnode,*head;  
public:sll()  
{  
    head=new node;  
    head->link=NULL;  
    head->data=NULL;  
}  
    void insert_end();  
    void create();  
    void getdata();  
    void display();  
};
```

```
void sll::create()  
{  
    ptr=new node;  
    cout<<"enter the no terminate -1 \n";  
    cin>>ptr->data;  
    head->link=ptr;  
    while(ptr->data!=-1)  
    {  
        ptr1=ptr;  
        ptr=new node;  
        cin>>ptr->data;  
        ptr1->link=ptr;  
    }  
    ptr1->link=NULL;
```

```

}
void sll::getdata()
{
    newnode=new node;
    cout<<"Enter the data to insert \n",
    cin>>newnode->data;
}
void sll::insert_end()
{
    ptr=head->link;
    while(ptr->link!=NULL)
        ptr=ptr->link;
    ptr->link=newnode;
    ptr=newnode;
    newnode->link=NULL;
}
void sll::display()
{
    ptr=head->link;
    cout<<"header";
    while(ptr!=NULL)
    {
        cout<<"->"<<ptr->data;
        ptr=ptr->link;
    }
    cout<<"->>null \n";
}
void main()
{
    sll s;
    clrscr();
    s.create();
    s.getdata();
    s.insert_end();
    s.display();
    getch();
}

```

----- OUTPUT-----

Enter the no terminate bu -1

22

33

44

55

66

-1

Enter the data to insert 100

header->22->33->44->55->66->100->null



```
/*17.PROGRAM TO SEARCH AN ELEMENT IN SINGLE LINKED LIST*/
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
struct node
```

```
{
```

```
    int data;
```

```
    node *link;
```

```
};
```

```
class sll
```

```
{
```

```
    node *ptr,*ptr1,*newnode,*head;
```

```
public:sll()
```

```
{
```

```
    head=new node;
```

```
    head->link=NULL;
```

```
    head->data=NULL;
```

```
}
```

```
    void search();
```

```
    void create();
```

```
    void display();
```

```
};
```

```
void sll::create()
```

```
{
```

```
    ptr=new node;
```

```
    cout<<"-----OUTPUT-----\n"<<endl;
```

```
    cout<<"enter the no terminate bu -1 \n";
```

```
    cin>>ptr->data;
```

```
    head->link=ptr;
```

```
    while(ptr->data!=-1)
```

```
    {
```

```
        ptr1=ptr;
```

```
        ptr=new node;
```

```
        cin>>ptr->data;
```

```
        ptr1->link=ptr;
```

```
    }
```

```
    ptr1->link=NULL;
```

```
}
```

```

void sll::search()
{
    int key;
    cout<<"Enter the element to search \n ";
    cin>>key;
    ptr=head->link;
    int flag=0;
    ptr1=NULL;
do
{
    if(ptr->data==key)
    {
        flag=1;
        ptr1=ptr;
    }
    else
        ptr=ptr->link;
}while(ptr!=NULL && flag==0);
if(flag==1)
cout<<"search is successfull"<< endl;
else
cout<<"search is unsuccessfull"<<endl;
}
void sll::display()
{
    ptr=head->link;
    cout<<"HEADER";
while(ptr!=NULL)
{
    cout<<"->"<<ptr->data;
    ptr=ptr->link;
}
    cout<<"->NULL";
}
void main()
{
    sll s;
    clrscr();
    s.create();
}

```



```
s.search();  
s.display();  
getch();  
}
```

-----OUTPUT-----

```
enter the no terminate bu -1  
33  
21  
77  
88  
6  
55  
3  
-1  
Enter the element to search 6  
search is successfull  
header->33->21->77->88->6->55->3->null
```

-----OUTPUT-----

```
enter the no terminate bu -1  
44  
33  
2  
1  
4  
5  
-1  
Enter the element to search 100  
search is unsuccessfull  
header->44->33->2->1->4->5->null
```

```
/* 18.PROGRAM TO DELET AN ELEMENT AT END POSITION IN  
DOUBLE LINKED LIST*/
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
struct node
```

```
{
```

```
    int data;
```

```
    node *llink,*rlink;
```

```
};
```

```
class dll
```

```
{
```

```
    node *ptr,*ptr1,*newnode,*head;
```

```
public:dll()
```

```
{
```

```
    head=new node;
```

```
    head->rlink=head;
```

```
    head->llink=head;
```

```
    head->data=NULL;
```

```
}
```

```
    void delet_end();
```

```
    void create();
```

```
    void getdata();
```

```
    void display();
```

```
};
```

```
void dll::create()
```

```
{
```

```
    ptr=new node;
```

```
    cout<<"-----OUTPUT-----\n";
```

```
    cout<<"enter the no teminate bu -1";
```

```
    cin>>ptr->data;
```

```
    head->rlink=ptr;
```

```
    head->llink=head;
```

```
while(ptr->data!=-1)
```

```
{
```

```
    ptr1=ptr;
```

```
    ptr=new node;
```

```
    cin>>ptr->data;
```

```

        ptr1->rlink=ptr;
        ptr->llink=ptr1;
    }
    ptr1->rlink=NULL;
}
void dll::getdata()
{
    newnode=new node;
    cout<<"Enter the data to Delete";
    cin>>newnode->data;
}
void dll::delet_end()
{
    ptr=head->rlink;
    while(ptr->rlink!=NULL)
    {
        ptr1=ptr;
        ptr=ptr->rlink;
    }
    ptr1->rlink=NULL;
}
void dll::display()
{
    ptr=head->rlink;
    cout<<"header";
    while(ptr!=NULL)
    {
        cout<<"->"<<ptr->data;
        ptr=ptr->rlink;
    }
    cout<<"->>null";
}

void main()
{
    dll s;
    clrscr();
    s.create();
}

```

```
s.getdata();
s.delet_end();
s.display();
}
```

-----OUTPUT-----

enter the no terminate bu -1

1

2

3

4

5

6

-1

Enter the data to Delete

6

header->1->2->3->4->5->null



```
/* 19.PROGRAM TO SORT THE GIVEN ELEMENTS IN DOUBLE LINKED  
LIST*/
```

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
struct node
```

```
{
```

```
    int data;
```

```
    node *llink,*rlink;
```

```
};
```

```
class dll
```

```
{
```

```
    node *ptr,*ptr1,*newnode,*head;
```

```
public:dll()
```

```
{
```

```
    head=new node;
```

```
    head->rlink=head;
```

```
    head->llink=head;
```

```
    head->data=NULL;
```

```
}
```

```
    void sort();
```

```
    void create();
```

```
    void getdata();
```

```
    void display();
```

```
};
```

```
void dll::create()
```

```
{
```

```
    ptr=new node;
```

```
    cout<<"-----OUTPUT----- \n"<<endl;
```

```
    cout<<"enter the no terminate bu -1 \n";
```

```
    cin>>ptr->data;
```

```
    head->rlink=ptr;
```

```
    while(ptr->data!=-1)
```

```
    {
```

```
        ptr1=ptr;
```

```
        ptr=new node;
```

```
        cin>>ptr->data;
```

```
        ptr1->rlink=ptr;
```

```

        }
        ptr1->rlink=NULL;
    }
void dll::sort()
{
    int temp;
    ptr=head->rlink;
    ptr1=ptr->rlink;
    while(ptr!=0)
    {
        while(ptr1!=0)
        {
            if(ptr->data>ptr1->data)
            {
                temp=ptr->data;
                ptr->data=ptr1->data;
                ptr1->data=temp;
            }
            ptr1=ptr1->rlink;
        }
        ptr1=ptr->rlink;
        ptr=ptr->rlink;
    }
    cout<<"The list after sorting \n"<<endl;
}
void dll::display()
{
    ptr=head->rlink;
    cout<<"HEADER";
    while(ptr!=NULL)
    {
        cout<<"->"<<ptr->data;
        ptr=ptr->rlink;
    }
    cout<<"->NULL";
}
void main()
{
    dll s;

```

```
clrscr();
s.create();
s.sort();
s.display();
getch();
}
```

-----OUTPUT-----

enter the no terminate bu -1

3

2

6

4

1

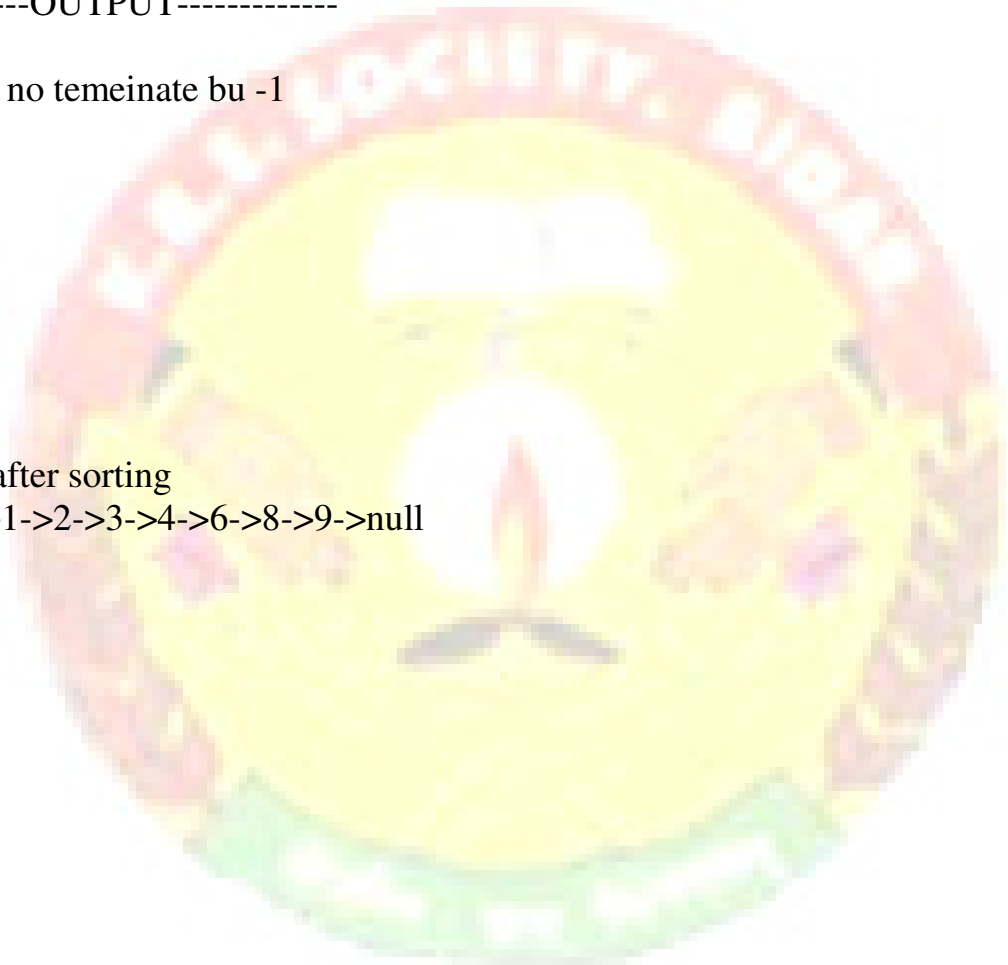
8

9

-1

The list after sorting

header->1->2->3->4->6->8->9->null



20. PROGRAM TO PRIFIX ORDER TRAVERSAL USING BINARY TREE

```
#include<iostream.h>
#include<conio.h>
#include<malloc.h>

struct node
{
    char info;
    struct node *left,*rt;
};
class btree
{
    node *bt;
public:
    node *create(node *);
    void pre(node *);
    void post(node *);
    void get();
};
void btree::get()
{
    int ch;
    cout<<"-----OUTPUT-----";
    cout<<"\n1.prefix\t\t2.postfix\n";
    cout<<"Enter the choice\n";
    cin>>ch;
    cout<<"\nEnter the root of a tree \n";
    bt=create(bt);
    switch(ch)
    {
        case 1:cout<<"\nprefix order traversal of binary tree\n";
                pre(bt);
                break;
        case 2:cout<<"\nPostfix order traversal of binary tree\n";
                post(bt);
                break;
    }
}
```



```

node *btree::create(node *ptr)
{
    char data;
    cin>>data;
    if(data!='0')
    {
        ptr=new node;
        ptr->info=data;
        ptr->left=ptr->rt=0;
        cout<<"\nEnter the left child "<<ptr->info<<":";
        ptr->left=create(ptr->left);
        cout<<"\nEnter the right child "<<ptr->info<<":";
        ptr->rt=create(ptr->rt);
    }return(ptr);
};

```

```

void btree::post(node *ptr)
{
    if(ptr!=0)
    {
        pre(ptr->left);
        pre(ptr->rt);
        cout<<ptr->info<<" ";
    }
}

```

```

void btree::pre(node *ptr)
{
    if(ptr!=0)
    {
        cout<<ptr->info<<" ";
        post(ptr->left);
        post(ptr->rt);
    }
}

```

```
void main()
{
    btree p;
    clrscr();
    p.get();
    getch();
}
```

-----OUTPUT-----

1.prefix 2.postfix

Enter the choice

1

Enter the root of a tree

a

Enter the left child a:b

Enter the left child b:0

Enter the right child b:0

Enter the right child a:c

Enter the left child c:0

Enter the right child c:0

prefix order traversal of binary tree

a b c