

```
#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<<"\nEnter n";
    cin>>n;
    cout<<"\nEnter Array Element:";
    for(i=0;i<n;i++)
        cin>>a[i];
}

void demo::insert()
{
    cout<<"\nEnter Position:";
    cin>>k;
    cout<<"\nEnter Item:";
    cin>>item;
    j=n;
    while(j>=k)
    {
        a[j+1]=a[j];
        j--;
    }
    a[k]=item;
    n++;
}

void demo::del()
{
    cout<<"\nEnter 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=0;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();
}

```

\*/ Output \*/

Enter n 3

Enter Array Element:1 2 4

1. Insert 2.Del 3.Dis 4. Exit

Enter choice 3

Elements are

1        2        4

Enter choice 1

Enter Position: 2

Enter Item: 6

Enter choice 3

Elements are

1        6        2        4

Enter choice 2

Enter Position: 3

Enter choice 3

Elements are

1        6        4

Enter choice 4

```
#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 trans();
        void mul();

};

void matrix::get()
{
    cout<<"\nEnter Number of Row & Column :\t";
    cin>>n>>m;

    cout<<"\nEnter the first Matrix:\n";

    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cin>>a[i][j];
    }
    cout<<"\nEnter Number of Row & Column :\t";
    cin>>p>>q;

    cout<<"\nEnter the first 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<<"\nThe 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<<"\nThe 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 matrix::trans()
{
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            e[i][j]=a[j][i];
        }
    }
    cout<<"\nThe Transpose of first matrix is :\n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cout<<e[i][j]<<"\t";
        cout<<"\n";
    }
}

void matrix::mul()
{
    if(m==p)
    {
        for(i=0;i<n;i++)
        {
            for(j=0;j<q;j++)
            {
                c[i][j]=0;
                for(k=0;k<p;k++)
                    c[i][j]=c[i][j]+a[i][k]*b[k][j];
            }
        }
    }
    cout<<"\nThe Matrix Multiplication is : \n";
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            cout<<c[i][j]<<"\t";
        cout<<"\n";
    }
}
else
    cout<<"\n Matrix Multiplication not possible";
}

```

```

}

void main()
{
    clrscr();
    matrix m;
    m.get();
    m.add();
    m.sub();
    m.trans();
    m.mul();
    getch();
}

*/ Output */

Enter Number of Row & Column :  3 3

Enter the first Matrix:
1 2 3
4 5 6
7 8 9

Enter Number of Row & Column :  3 3

Enter the first Matrix:
1 2 3
4 5 6
7 8 9

The addition of two matrix is :
2      4      6
8      10     12
14     16     18

The Substraction of two matrix is :
0      0      0
0      0      0
0      0      0

The Transpose of first matrix is :
1      4      7
2      5      8
3      6      9

The Matrix Multiplication is :
30     36     42
66     81     96
102    126    150

```

```
#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<<"\nStack is overflow:";
    else
    {
        cout<<"\nEnter element:";
        cin>>ele;
        top++;
        s[top]=ele;
    }
}

void stack::dis()
{
    cout<<"\nElements 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<<"\nEnter element:";
        cin>>n;
        s[top-i+1]=n;
    }
}

void main()
{
    clrscr();
    stack s;
    int ch;
    cout<<"\n1. 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();
}

```

\*/ Output \*/

1. Push 2.Display 3.Pop 4.Peep 5.Change 6.Exit

Enter ch :1

Enter element:10

Enter ch :1

```
Enter element:20

Enter ch :1

Enter element:30

Enter ch :1

Stack is overflow:
Enter ch :2

Elements in stack are:
30      20      10
Enter ch :3

Pop ele is 30
Enter ch :2

Elements in stack are:
20      10
Enter ch :4

Enter position:1

Peep ele is 20
Enter ch :
2

Elements in stack are:
20      10
Enter ch :5

Enter position 1

Enter element:80

Enter ch :2

Elements in stack are:
80      10
Enter ch : 6
```



```
#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 element 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<<"\nEnter 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();
}

*/ Output */

```

1.insert 2.display 3.delete 4. exit

Enter ch:3

Underflow  
Enter ch:1

Enter n10

Enter ch:1

Enter n20

Enter ch:1

Enter n30

Enter ch:1

Overflow  
Enter ch:2

Elements in queue are:10                      20                      30  
Enter ch:3

Deleted element is 10  
Enter ch:2

Elements in queue are:20                      30  
Enter ch:4

-----  
Assignment Name: Perform Insert, Display, delete, search, sum operation  
on LL

Class: BCA-II

Lab: BCA 407(DS)  
-----

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class node
{
    int info,item,s;
    node *link;
public:
    void insert();
    void dis();
    void del();
    void search();
    void sum();
};
node *move,*start=NULL,*temp;

void node::insert()
{
    cout<<"\nEnter the item:";
    cin>>item;
    node *nodel=new node;
    nodel->link=NULL;
    nodel->info=item;
    if(start==NULL)
        start=nodel;
    else
    {
        move=start;
        while(move->link!=NULL)
            move=move->link;
        move->link=nodel;
    }
}

void node::dis()
{
    node *x;
    x=start;
    cout<<"\n Elements in LL are:";
    while(x!=NULL)
    {
        cout<<"\t"<<x->info;
        x=x->link;
    }
}

void node::sum()
{
    node *x;
    x=start;
    s=0;
    while(x!=NULL)
    {
        s=s+x->info;
```

```

        x=x->link;
    }
    cout<<"\nSum of node is"<<s;
}

void node::del()
{
    temp=start;
    if(temp!=NULL)
    {
        temp=temp->link;
        cout<<"\nDeleted node is"<<start->info;
        start=temp;
    }
    else
        cout<<"\n List is empty:";
}

void node::search()
{
    int c=0,f=0,d;
    cout<<"\nEnter item";
    cin>>item;
    temp=start;
    while(temp!=NULL)
    {
        c++;
        if(temp->info==item)
        {
            f=1;
            d=c;
            break;
        }
        temp=temp->link;
    }
    if(f==1)
        cout<<"\nElement is found at position "<<d;
    else
        cout<<"\nElement is not found";
}

void main()
{
    clrscr();
    node n;
    int ch;
    cout<<"\n1.Insert  2.Display 3. Delete 4.Search 5.Sum 6.Exit\n";

    do
    {
        cout<<"\nEnter choice";
        cin>>ch;
        switch(ch)
        {
            case 1: n.insert(); break;
            case 2: n.dis(); break;
            case 3: n.del(); break;
            case 4: n.search(); break;
            case 5: n.sum(); break;
            case 6: exit(0);
        }
    }
}

```

```

        }while(ch!=6);
        getch();
    }

    /* Output */

    1.Insert  2.Display 3. Delete 4.Search 5.Sum 6.Exit

    Enter choice1

    Enter the item:10

    Enter choice1

    Enter the item:20

    Enter choice1

    Enter the item:30

    Enter choice2

    Elements in LL are:    10        20        30
    Enter choice3

    Deleted node is10
    Enter choice2

    Elements in LL are:    20        30
    Enter choice5

    Sum of node is50
    Enter choice4

    Enter item30

    Element is found at position 2
    Enter choice4

    Enter item19

    Element is not found
    Enter choice 6

```



```

        f=1;
    }
    while (temp!=NULL)
    {
        c++;
        p=temp;
        temp=temp->link;
        if (c==pos-1)
        {
            f=1;
            p->link=temp->link;
        }
    }
    if (f==0)
        cout<<"\n node is not found";
}

void node::del_info()
{
    int pos, f=0;
    node *p;
    cout<<"\nEnter the element:";
    cin>>item;
    temp=start;
    if (start==NULL)
        cout<<"\nLL is Empty:";
    if (start->info==item)
    {
        start=start->link;
        f=1;
    }
    while (temp!=NULL)
    {
        p=temp;
        temp=temp->link;
        if (temp->info==item)
        {
            f=1;
            p->link=temp->link;
        }
    }
    if (f==0)
        cout<<"\n node is not found";
}

void main()
{
    clrscr();
    node n;
    int ch;
    cout<<"\n1.Insert 2.Display 3.Del_position 4.Del_information 5.exit:\n";
    while (ch!=5)
    {
        cout<<"\nEnter choice";
        cin>>ch;
        switch (ch)
        {
            case 1: n.insert(); break;
            case 2: n.dis(); break;
            case 3: n.del_pos(); break;
            case 4: n.del_info(); break;

```

```

        case 5: exit(0);
    }
}
getch();
}

*/ Output */

1.Insert 2.Display 3.Del_position 4.Del_information 5.exit:
Enter choice1
Enter the item:10
Enter choice1
Enter the item:20
Enter choice1
Enter the item:-3
Enter choice2
    10      20      -3
Enter choice3
Enter Position:2
Enter choice2
    10      -3
Enter choice4
Enter the element:-3
Enter choice2
    10
Enter choice 5

```



-----  
Assignment Name: Implement Doubly Link List  
Class: BCA-II

Lab: BCA 407(DS)  
-----

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class node
{
    int info,c,j;
    node *left,*right;
public:
    void insert();
    void display();
    void del();
};

node *start=NULL,*temp=NULL,*move=NULL, *temp1=NULL;
void node::insert()
{
    int item;
    node *p=new node;
    cout<<"\nEnter element:";
    cin>>item;
    p->info=item;
    p->left=NULL;
    p->right=NULL;
    if(start==NULL)
    {
        start=p;
        return;
    }
    else
    {
        temp=start;
        while(temp->right!=NULL)
            temp=temp->right;
        temp->right=p;
        p->left=start;
    }
}

void node::display()
{
    move=start;
    if(move==NULL)
    {
        cout<<"\n LL Empty:";
        return;
    }
    else
    {
        cout<<"\n node in DLL are :";
        while(move!=NULL)
        {
            cout<<move->info<<"\t";
            move=move->right;
        }
    }
}
```

```

}

void node::del()
{
    if(start==NULL)
    {
        cout<<"\n LL Empty:";
        return;
    }
    temp=start;
    start=temp->right;
    start->left=NULL;
    temp->right=NULL;
    cout<<"\n deleted element is"<<temp->info;
}

void main()
{
    clrscr();
    node n;
    int ch;
    cout<<"\n1. Insert 2. Display 3.Delete 4. Exit";
    while(ch!=4)
    {
        cout<<"\nEnter choice";
        cin>>ch;
        switch(ch)
        {
            case 1: n.insert(); break;
            case 2: n.display(); break;
            case 3: n.del(); break;
            case 4: exit(0);
        }
    }
    getch();
}

```

\*/ Output \*/

```

1. Insert 2. Display 3.Delete 4. Exit
Enter choice2

```

```

    LL Empty:
Enter choice1

```

```

Enter element:10

```

```

Enter choice1

```

```

Enter element:20

```

```

Enter choice1

```

```

Enter element:30

```

```

Enter choice2

```

```

    node in DLL are :10    20    30
Enter choice3

```

deleted element is10  
Enter choice2

node in DLL are :20 30  
Enter choice3

deleted element is20  
Enter choice3

deleted element is30  
Enter choice2

LL Empty:  
Enter choice3

LL Empty:  
Enter choice

```
#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<<"\nEnter the array element:";
    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;
        }
    }
}
```

```

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

    if(exch==0)
        return;
    else
        last=last-1;
}

void demo::disp()
{
    cout<<"\nThe array element are";
    for(i=1;i<=n;i++)
        cout<<a[i]<<"\t";
}

void main()
{
    clrscr();
    demo d;
    d.get();
    d.disp();
    d.asc_sort();
    cout<<"\nAfter Ascending Sort:";
    d.disp();
    d.dec_sort();
    cout<<"\nAfter Descending Sort:";
    d.disp();
    getch();
}

*/ Output */

```

Enter the array size: 3

Enter the array element: 12 3 45

The array element are12 3            45

After Ascending Sort:

The array element are3 12            45

After Descending Sort:

The array element are45 12           3

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

class demo
{
    int a[10],i, min_index,j,n,temp;
public:
    void get();
    void asc_sort();
    void dsc_sort();
    void disp();
};

void demo::get()
{
    cout<<"\nEnter the array size:";
    cin>>n;
    cout<<"\nEnter the array element:";
    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++)
    {
        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::disp()
{
    cout<<"\n The array element are";
    for(i=1;i<=n;i++)
        cout<<a[i]<<"\t";
}

void main()
{
    clrscr();
    demo d;
    d.get();
    d.disp();
    d.asc_sort();
    cout<<"\nAfter ascending sort:";
    d.disp();
    d.dsc_sort();
    cout<<"\n After Descending sort:";
    d.disp();
    getch();
}

*/ Output */

Enter the array size:4

Enter the array element:12 3 -45 -6

The array element are12      3      -45      -6
After ascending sort:
The array element are-45      -6      3      12
After Descending sort:
The array element are12      3      -6      -45

```

```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
class insert
{
    int n,a[10],temp,ptr,q,i,j,k,key;
public:
    void get();
    void sort();
    void display();
};

void insert::get()
{
    cout<<"\nEnter Range:";
    cin>>n;
    for(i=1;i<=n;i++)
        a[i]=random(1000);
    cout<<"\nElements are :";
    for(i=1;i<=n;i++)
        cout<<a[i]<<"\t";
}

void insert::sort()
{
    a[0]=-9999;
    for(i=2;i<=n;i++)
    {
        temp=a[i];
        ptr=i-1;
        while(temp<a[ptr])
        {
            a[ptr+1]=a[ptr];
            ptr--;
        }
        a[ptr+1]=temp;
    }
}

void insert::display()
{
    cout<<"\nSorted Element using Insertion Sort:";
    for(i=1;i<=n;i++)
        cout<<a[i]<<"\t";
}

void main()
{
    clrscr();
    insert h;
    h.get();
    h.sort();
    h.display();
    getch();
}
```

\*/ Output \*/

Enter Range:5

Elements are :10            3            335            33            355

Sorted Element using Insertion Sort:3    10            33            335            355



-----  
Assignment Name: Implement Linear and Binary Search

Class: MCA I

Lab: CA Lab III (DS)  
-----

```
#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<<"\nEnter array Elements:";
    for(i=1;i<=n;i++)
        cin>>a[i];
}

void demo::linear()
{
    int ele;
    cout<<"\nEnter the element to be search";
    cin>>ele;
    for(i=1;i<=n;i++)
    {
        if(a[i]==ele)
        {
            cout<<"\nSuccessful search";
            cout<<"\nElement is found at position "<<i;
            return;
        }
    }
    if(i>n)
    {
        cout<<"\nUnsuccessful search:";
        cout<<"\nElement 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<<"\n Element 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<<"\nEnter 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();
}

```

```
*/ Output */

Enter n:3

Enter array Elements:12 3 45

Element are
12      3      45
1:Linear 2:Binary 3:exit

Enter Choice:1

Enter the element to be search 3

Successful search
Element is found at position 2
Enter Choice:2

Element are
45      12      3
Enter element to be search 12

Element is found at :2
Enter Choice:2

Element are
45      12      3
Enter element to be search 56

Element is not found:
Enter Choice:3
```

-----  
Assignment Name: Implement Tower of Hanoi

Class: BCA-II

Lab: BCA 407(DS)  
-----

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

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

```
class demo
```

```
{
```

```
    int n;
```

```
public:
```

```
    void tower(int,char,char,char);
```

```
    void get();
```

```
};
```

```
void demo::get()
```

```
{
```

```
    cout<<"\nEnter the number of disk: ";
```

```
    cin>>n;
```

```
    tower(n, 'A', 'B', 'C');
```

```
}
```

```
void demo::tower(int n,char beg,char aux,char end)
```

```
{
```

```
    if(n!=0)
```

```
    {
```

```
        tower(n-1,beg,end,aux);
```

```
        cout<<"\n Move disk "<<n<<" from "<<beg<<" to "<<end<<"\n";
```

```
        tower(n-1,aux,beg,end);
```

```
    }
```

```
}
```

```
void main()
```

```
{
```

```
    clrscr();
```

```
    demo d;
```

```
    d.get();
```

```
    getch();
```

```
}
```

```
*/ Output */
```

```
Enter the number of disk: 3
```

```
Move disk 1 from A to C
```

```
Move disk 2 from A to B
```

```
Move disk 1 from C to B
```

```
Move disk 3 from A to C
```

```
Move disk 1 from B to A
```

```
Move disk 2 from B to C
```

```
Move disk 1 from A to C
```

```
#include<iostream.h>
#include<conio.h>
class factorial
{
    double n,f;
public:
    void get();
    double fact(double);
};

void factorial::get()
{
    cout<<"\nEnter n";
    cin>>n;
    f=fact(n);
    cout<<"\n Factorial of "<<n<<" is "<< f;
}

double factorial::fact(double n)
{
    if(n==0)
        return 1;
    else
        return(n*fact(n-1));
}

void main()
{
    clrscr();
    factorial f;
    f.get();
    getch();
}
```

\*/ Output \*/

Enter n 34

Factorial of 34 is 2.952328e+38