```
--------------------------------------------------------------------------
Assignment Name: Demonstration of Array
Class: MCA I
--------------------------------------------------------------------------

#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=1;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=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();
}

*/ Output */

Enter no 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
```

---

**Assignment Name: Demonstration of Matrix**
**Class: MCA I**

---

```
#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<<"\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 main()
{
clrscr();
matrix m;
m.get();
m.add();
m.sub();
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
```

---

**Assignment Name: Implement Stack for Integer**
**Class: MCA I**

---

```
#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
```

```
--------------------------------------------------------------------------
Assignment Name: Implement Infix to Postfix
Class: MCA I
--------------------------------------------------------------------------
#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();
}

*/ Output */


Enter infix Expression:(a*b-(c+d/e^f)*h)

Postfix Expression is :ab*cdef^/+h*-

Enter infix Expression:a+2*5

Postfix Expression is :a25*+
```

```
--------------------------------------------------------------------------
Assignment Name: Implement linear queue for integer
Class: MCA I
--------------------------------------------------------------------------

#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: Implement Circular Queue for integer
Class: MCA I
--------------------------------------------------------------------------


#include<iostream.h>
#include<conio.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<<"\n Overflow";
     }
     else
     {
          if(r==4)
          r=-1;
          r++;
          cout<<"\nEnter 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);
}


*/ Output */

1: Push 2: Pop 3:show 4:exit
Enter choice1

 Overflow
 1: Push 2: Pop 3:show 4:exit
Enter choice3

10
20
30
40
50
 1: Push 2: Pop 3:show 4:exit
Enter choice2

Deleted element is :10
 1: Push 2: Pop 3:show 4:exit
Enter choice2

Deleted element is :20
 1: Push 2: Pop 3:show 4:exit
Enter choice3

30
40
50
 1: Push 2: Pop 3:show 4:exit
Enter choice1

Enter item :44

 1: Push 2: Pop 3:show 4:exit
Enter choice1

Enter item :55

 1: Push 2: Pop 3:show 4:exit
Enter choice1

 Overflow
 1: Push 2: Pop 3:show 4:exit
Enter choice3
```

```
30
40
50
44
55
 1: Push 2: Pop 3:show 4:exit
Enter choice 4
```

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

#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 *node1=new node;
     node1->link=NULL;
     node1->info=item;
     if(start==NULL)
          start=node1;
     else
     {
          move=start;
          while(move->link!=NULL)
          move=move->link;
          move->link=node1;
     }
}

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
```

```
--------------------------------------------------------------------------
Assignment Name: Program to Implement Stack using LL
--------------------------------------------------------------------------

#include<conio.h>
#include<iostream.h>
#include<process.h>
class stack
{
     int info, ele;
     stack *node,*link,*top;
public:
     stack()
     {
          top=NULL;
     }
     void insert();
     void del();
     void dis();
};

void stack::insert()
{
     node=new stack;
     cout<<"\nEnter Info:";
     cin>>ele;
     node->info=ele;
     node->link=NULL;
     if(top==NULL)
     {
          top=node;
     }
     else
     {
          node->link=top;
          top=node;
     }
}
void stack::del()
{
     if(top==NULL)
     {
          cout<<"\n Underflow";
     }
     else
     {
          cout<<"\nDeleted Element is :"<<top->info;
          top=top->link;
     }
}

void stack::dis()
{
     stack *move;
```

```
      move=top;
      while(move!=NULL)
      {
            cout<<"\t"<<move->info;
            move=move->link;
      }
}

void main()
{
      clrscr();
      int ch;
      stack s;
      cout<<"\n1.Insert 2.Show 3.Delete 4.Exit";
      while(ch!=4)
      {
            cout<<"\nEnter Choice";
            cin>>ch;
            switch(ch)
            {
                  case 1: s.insert(); break;
                  case 2: s.dis(); break;
                  case 3: s.del(); break;
                  case 4:exit(0);
            }
      }
getch();
}

*/ Output */

1.Insert 2.Show 3.Delete 4.Exit
Enter Choice1

Enter Info:23

Enter Choice1

Enter Info:55

Enter Choice1

Enter Info:66

Enter Choice1

Enter Info:77

Enter Choice2
        77      66      55      23
Enter Choice3
Deleted Element is :77
Enter Choice2
        66      55      23
```

```
--------------------------------------------------------------------------
Assignment Name: Perform Deletion in LL according to position &
                 information
--------------------------------------------------------------------------
#include<iostream.h>
#include<conio.h>
#include<process.h>

class node
{
     int info,item;
     node *link;
public:
     void insert();
     void dis();
     void del_info();
     void del_pos();
};
node *move,*start,*temp;

void node::insert()
{
     cout<<"\nEnter the item:";
     cin>>item;
     node *node1=new node;
     node1->link=NULL;
     node1->info=item;
     if(start==NULL)
          start=node1;
     else
     {
          move=start;
          while(move->link!=NULL)
          move=move->link;
          move->link=node1;
     }
}
void node::dis()
{
     node *x;
     x=start;
     while(x!=NULL)
     {
          cout<<"\t"<<x->info;
          x=x->link;
     }
}

void node::del_pos()
{
     int pos,f=0,c=0;
     node *p;
     cout<<"\nEnter Position:";
     cin>>pos;
```

```
      temp=start;
      if(start==NULL)
            cout<<"\nLL is empty\n";
      if(pos==1)
      {
            start=start->link;
            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: Perform Sort on LL
Class: MCA I
--------------------------------------------------------------------------

#include<iostream.h>
#include<conio.h>
#include<process.h>
class node
{
     int info,item,s;
     node *link;
public:
     void insert();
     void sort();
     void dis();
};

node *move,*start=NULL, *temp;

void node::insert()
{
     cout<<"\nEnter the item";
     cin>>item;
     node *node1=new node;
     node1->info=item;
     node1->link=NULL;
     if(start!=NULL)
          node1->link=start;
          start=node1;
}

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

void node::sort()
{
     node *t=start;
     int c=0,j;
     while(t!=NULL)
     {
          c++;
          t=t->link;
```

```
      }

      for(j=1;j<=c;j++)
       for(t=start;t->link!=NULL;t=t->link)
           if((t->info)>(t->link)->info)
           {
                int a;
                a=t->info;
                t->info=(t->link)->info;
                (t->link)->info=a;
           }
      cout<<"\nAfter Sorting: ";
}

void main()
{
      clrscr();
      node n;
      int ch;
      cout<<"\n1.Insert 2.Display 3. Sort 4.Exit\n";
      while(ch!=4)
      {
           cout<<"\n Enter Choice\n";
           cin>>ch;
           switch(ch)
           {
                case 1: n.insert(); break;
                case 2: n.dis(); break;
                case 3: n.sort(); break;
                case 4: exit(0);
           }
      }
      getch();
}

*/ Output */

1.Insert 2.Display 3. Sort 4.Exit

 Enter Choice
1

Enter the item10

 Enter Choice
1

Enter the item-2

 Enter Choice
1

Enter the item-1
```

```
 Enter Choice
1

Enter the item4

 Enter Choice
2

 Element in LL are:     4       -1      -2      10
 Enter Choice
3

After Sorting:
 Enter Choice
2

 Element in LL are:     -2      -1      4       10
```

```
--------------------------------------------------------------------------
Assignment Name: Implement Reverse on LL
Class: MCA I
--------------------------------------------------------------------------

#include<iostream.h>
#include<process.h>
#include<conio.h>
class node
{
     int info;
     node *link;
public:
     void insert();
     void dis();
     void reverse();
};

node *move=NULL,*start=NULL,*temp=NULL;

void node::insert()
{
     int item;
     cout<<"\nEnter item:";
     cin>>item;
     node *node1=new node;
     node1->link=NULL;
     node1->info=item;
     if(start==NULL)
          start=node1;
     else
     {
          move=start;
          while(move->link!=NULL)
          move=move->link;
          move->link=node1;
     }
}
void node::dis()
{
     node *x;
     x=start;
     while(x!=NULL)
     {
          cout<<"\t"<<x->info;
          x=x->link;
     }
}

void node::reverse()
{
```

```
      node *temp1,*temp2;
      temp=start;
      temp1=temp->link;
      temp2=temp1->link;
      temp->link=NULL;
      while(temp1!=NULL)
      {
            temp1->link=temp;
            temp=temp1;
            temp1=temp2;
            temp2=temp2->link;
      }
      start=temp;
      dis();
}


void main()
{
      clrscr();
      node n;
      int ch;
      cout<<"\n 1.Insert 2. Display 3.Reverse 4.Exit\n";
      while(ch!=4)
      {
            cout<<"\nEnte ch \n";
            cin>>ch;
            switch(ch)
            {
                  case 1: n.insert(); break;
                  case 2: n.dis(); break;
                  case 3: n.reverse(); break;
                  case 4: exit(0);
            }
      }
      getch();
}




*/ Output */


 1.Insert 2. Display 3.Reverse 4.Exit

Ente ch
1

Enter item:10

Ente ch
```

```
1

Enter item:20

Ente ch
1

Enter item:30

Ente ch
1

Enter item:40

Ente ch
2
        10      20      30      40
Ente ch
3
        40      30      20      10
Ente ch
4
```

```
--------------------------------------------------------------------------
Assignment Name: Perform Bubble Sort
Class: MCA I
--------------------------------------------------------------------------

#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;
            }
      }

      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
```

```
--------------------------------------------------------------------------
Assignment Name: Perform Selection Sort
Class: MCA I
--------------------------------------------------------------------------

#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
```

```
--------------------------------------------------------------------------
Assignment Name: Implement Insertion Sort
Class: MCA I
--------------------------------------------------------------------------

#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 Quick sort for integer in ascending order
Class: MCA I
--------------------------------------------------------------------------

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

class demo
{
     int x[20],temp;
     int a,n,i,j,left,right;
public:

     void get();
     void asort(int,int);
     int partition(int,int);
     void disp();
};


void demo::get()
{
     cout<<"\nEnter the array size:";
     cin>>n;
     cout<<"\nEnter the array element:";
     for(i=1;i<=n;i++)
     cin>>x[i];
     asort(1,n);
}

void demo::asort(int p,int q)
{
     if(p<q)
     {
          j=partition(p,q);
          asort(p,j-1);
          asort(j+1,q);
     }
}

int demo::partition(int lb, int ub)
{
     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<<"\nThe array element are:";
      for(i=1;i<=n;i++)
      cout<<x[i]<<"\t";
}

void main()
{
      clrscr();
      demo d;
      d.get();
      cout<<"\nAfter Ascending sort";
      d.disp();
      getch();
}

*/ Output */


Enter the array size: 5

Enter the array element:12 3 -45 -67 8

After Ascending sort
The array element are:-67       -45     3       8       12
```

```
--------------------------------------------------------------------------
Assignment Name: Implement Merge sort in ascending order
Class: MCA I
--------------------------------------------------------------------------

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

int n;  //remember that n should be declare global

class merge
{
     int a[10],b[10],i,j;
public:

     void read();
     void merge_sort(int l,int h);
     void merge1(int l,int m, int h);
     void disp();
};


void merge::read()
{
     for(i=0;i<n;i++)
          cin>>a[i];
}

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);
          merge1(l,mid,h);
     }
}

void merge::merge1(int low,int m,int high)
{

     int i=low;
     int k=low;
     j=m+1;

     while((i<=m)&&(j<=high))
     {
          if(a[i]<=a[j])//Change
          {
```

```
                b[k]=a[i];
                i++;
                k++;
          }
          else
          {
                b[k]=a[j];
                j++;
                k++;
          }
     }


          while(i<=m)
          {
                b[k]=a[i];
                i++;
                k++;
          }

          while(j<=high)
          {
                b[k]=a[j];
                j++;
                k++;
          }

     for(int k1=low;k1<=high;k1++)
          a[k1]=b[k1];
}

void merge::disp()
{
     for(i=0;i<n;i++)
     cout<<a[i]<<"\t";
}


void main()
{
     clrscr();
     int l,h;
     merge m;
     cout<<"\nEnter Elements";
     cin>>n;
     h=n-1;
     l=0;
          m.read();
          cout<<"\n\nDisplay the array elements\n";
          m.disp();
          m.merge_sort(l,h);
          cout<<"\nAfter Sorting\n";
          m.disp();
          getch();
```

```
}

*/ Output */

Enter Elements5
12 -34 5 67 -8

Display the array elements
12      -34     5       67      -8
After Sorting
-34     -8      5       12      67
```

```
--------------------------------------------------------------------------
Assignment Name: Implement Linear and Binary Search
Class: MCA I
--------------------------------------------------------------------------

#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
```