```
-----------------------------------------------------------------------------
Assignment Name: Program for array perform insert, delete & display operation.
Class: MCA I                                        Lab: CA LAB-IV (DS)
-----------------------------------------------------------------------------
```

```
#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 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
```

```
-----------------------------------------------------------------------------
Assignment Name: Program for matrix addition, substraction, multiplication and
transpose of matrix
Class: MCA I                                       Lab: CA LAB-IV (DS)
------------------------------------------------------------------------------
#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 Second Matrix:\n";

      for(i=0;i<p;i++)
      {
            for(j=0;j<q;j++)
            cin>>b[i][j];
      }
}

void matrix::add()
{
      cout<<"\nThe addition of two matrix is :\n";
      for(i=0;i<n;i++)
      {
            for(j=0;j<m;j++)
            {
                  c[i][j]=a[i][j]+b[i][j];
                  cout<<c[i][j]<<"\t";
            }
         cout<<"\n";

      }

}
```

```
void matrix::sub()
{
    cout<<"\nThe Subtraction of two matrix is :\n";
     for(i=0;i<n;i++)
      {
           for(j=0;j<m;j++)
           {
                 d[i][j]=a[i][j]-b[i][j];
                cout<<d[i][j]<<"\t";
           }
         cout<<"\n";

      }
}

void matrix::trans()
{
      cout<<"\nThe Transpose of first matrix is :\n";

      for(i=0;i<n;i++)
      {
           for(j=0;j<m;j++)
           {
                 e[i][j]=a[j][i];
                 cout<<e[i][j]<<"\t";

           }
           cout<<"\n";
      }
}

void matrix::mul()
{
      cout<<"\nThe Matrix Multiplication is : \n";

      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<<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
```

```
----------------------------------------------------------------------------
Assignment Name: Implement Stack for Integer/character perform different
operation on stack (push, pop, peep, change).
Class: MCA I                                       Lab: CA LAB-IV (DS)
----------------------------------------------------------------------------

#include<iostream.h>
#include<conio.h>
#include<process.h>
int n;
class stack
{
  private:
      int s[10],top,ele,i;  // char s[10] for character
  public:
      stack()
      {
           top=-1;
      }
      void push();
      void dis();
      void pop();
      void peep();
      void change();
};
void stack::push()
{
      if(top>=n-1)
       cout<<"\nStack is overflow:";
      else
      {
       cout<<"\nEnter element:";
       cin>>ele;
       top++;
       s[top]=ele;
      }
}

void stack::dis()
{
      if(top==-1)
      {
       cout<<"\n Stack is Empty";
      }
      else
      {
      cout<<"\nElements in stack are:\n";
      for(i=top;i>=0;i--)
       cout<<s[i]<<"\t";
      }
}
void stack::pop()
{
      if(top==-1)
      {
           cout<<"\nUnderflow";
      }
      else
        {
           cout<<"\nPop ele is "<<s[top];
                 top--;
       }
}

void stack::peep()
{
      cout<<"\nEnter position:";
      cin>>i;
      if((top-i+1)<0)
      {
            cout<<"\nUnderflow";
      }
      else
       {
      cout<<"\nPeep ele is "<<s[top-i+1];
      }
}

void stack::change()
{
      cout<<"\nEnter position ";
      cin>>i;
      if((top-i+1)<0)
      {
            cout<<"\nUnderflow";
      }
      else
      {
            int n;                //char n; for character
            cout<<"\nEnter element:";
            cin>>n;
            s[top-i+1]=n;
      }
}

void main()
{
      clrscr();
      stack s;
      cout<<"Enter size of stack";
      cin>>n;
      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: s.pop();break;
                  case 4: s.peep(); break;
                  case 5: s.change(); break;
                  case 6: exit(0);
            }
      }
      getch();
}
```

```
*/ Output */


Enter size of stack  3

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 operation using stack.
Class: MCA I                                       Lab: CA LAB-IV (DS)
-----------------------------------------------------------------------------
#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 / character perform
different operation on queue ( insert,delete,display)
Class: MCA I                                       Lab: CA LAB-IV (DS)
-------------------------------------------------------------------------------

#include<iostream.h>
#include<conio.h>
#include<process.h>
int m;
class queue
{
      int f,r,q[10],n,i;   //char q[10],n  for character
public:
      queue()
      {
            f=r=0;
      }
      void insert();
      void del();
      void dis();
};

void queue::insert()
{
      if(r==3)
       cout<<"\nOverflow";
      else
      {
       cout<<"\nEnter Element in Queue=";
       cin>>n;
       if(f==0)
          f=1;
          r++;
          q[r]=n;
      }
}

void queue::del()
{
      if(f==0)
      {
            cout<<"\nUnderflow";
      }
      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<<"Enter size of queue";
     cin>>m;
      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 */

Enter size of queue 3

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

Enter ch:3

Underflow
Enter ch:1

Enter Element in Queue=10
Enter ch:1

Enter Element in Queue=20
Enter ch:1

Enter Element in Queue=30
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, perform different operation of circular queue (push ,pop, show)

Class: MCA I   Lab: CA LAB-IV (DS)

```
#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
```