Computer Science Program File

January 14

2019

Roll No 9202624

Shreyas
Khandekar

INDEX

Acknowledgement

TO WHOM SO EVER IT MAY CONCERN

I convey my sincere gratitude to Mr Gautam Sarkar for his continuous guidance and help. Without his kind support, the completion of the project would not have been possible. I also convey my sincere thanks Ms Divya Sahdev for her support and guidance.

SHREYAS KHANDEKAR

CLASS S7D

BOARD ROLL NO.

Certificate

TO WHOM SO EVER IT MAY CONCERN

This is to certify that the practical work in this file has been designed and developed by Shreyas Khandekar under my guidance and supervision.

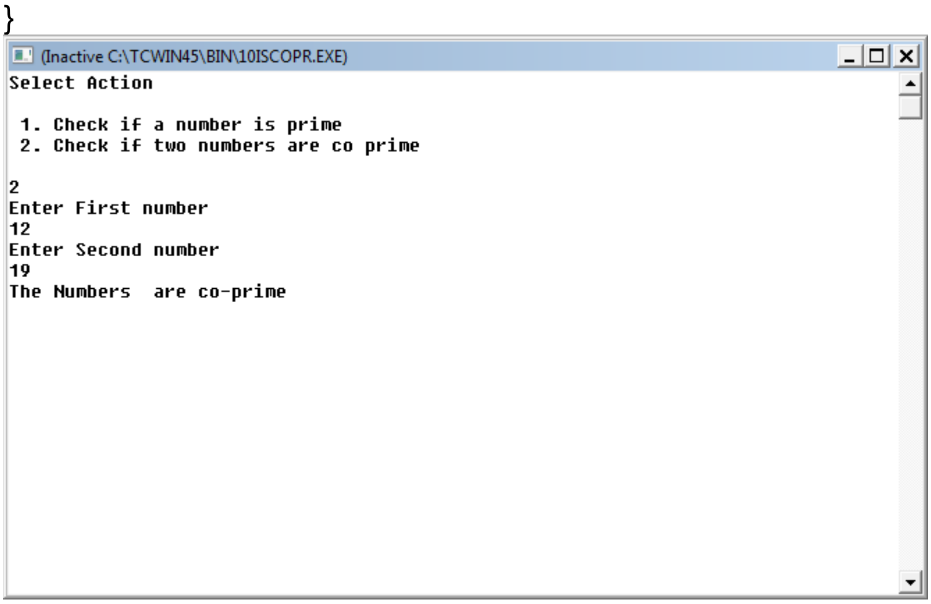
GAUTAM SARKAR

HEAD, COMPUTER DEPARTMENT

MODERN SCHOOL, BARAKHAMBA ROAD

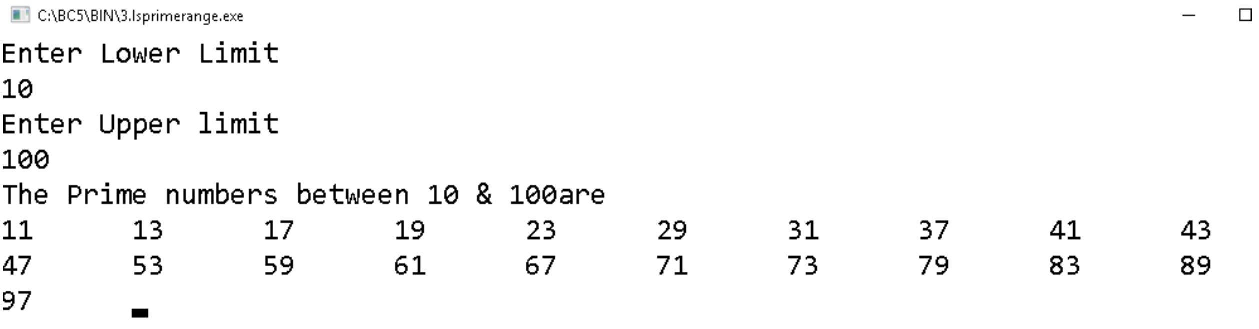
```
/* 1.
Define the following functions having same name PRIME( ) and also write the
required program to demonstrate their use.

(i) If one argument is passed, the function should check whether the number is a
prime and return 1 (if yes) or 0 (if not) accordingly.

(ii)        If two argument is passed, then the function should check whether the
numbers are co-prime or not and return 1 (if yes) or 0 (if not) accordingly.
*/


#include<iostream.h>
#include<conio.h>
#include<ctype.h>
#include<stdlib.h>
#include<math.h>
int isprime(int num)
{
   int flag=1;
   for(int factor=2;factor<=sqrt(num);factor++)
   {
           if(num%factor==0)
   flag=0;
   }

  return flag;
}

int isprime(int num1,int num2)
{
   int flag=1,num;
  if(num1>num2)
    num2=num;
  else
     num1=num;
  for(int factor=2;factor<=sqrt(num);factor++)
   {
           if(num1%factor==0&&num2%factor==0)
   flag=0;
   }

  return flag;
}


void main()
{
    char op;
  cout<<"Select Action"<<endl;
```

```
                                                    cout<<"\n 1. Check if a number is prime
                                                  \n 2. Check if two numbers are co  prime
                                                  \n"<<endl;
  cin>>op;
  switch(op)
  {
        case '1':   int x;              cout<<"Enter a number"<<endl;
                                        cin>>x;
                                 if(isprime(x)==1)
                                {
                                        cout<<"The Number is prime"<<endl;
                                }
                                else
                                {
                                        cout<<"The Number is not prime"<<endl;
                                }
                                break;
    case '2':                   cout<<"Enter First number"<<endl;
                                int a,b;
                                cin>>a;
                                 cout<<"Enter Second number"<<endl;
                                cin>>b;
                                if(isprime(a,b)==1)
                                {
                                        cout<<"The Numbers  are co-prime"<<endl;
                                }
                                else
                                {
                                        cout<<"The Numbers are not co-prime"<<endl;
                                }
                                        break;
    default     :               cout<<"Invalid Input";

 }
  getch();
}
```



```
Select Action

 1. Check if a number is prime
 2. Check if two numbers are co prime

2
Enter First number
12
Enter Second number
19
The Numbers  are co-prime
```

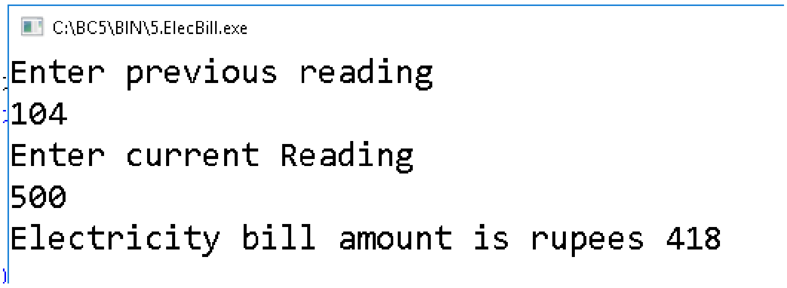
```
//2
// To Display all prime numbers in a user defined range
#include<iostream.h>
#include<conio.h>
#include<math.h>
int isprime(int num)
{
   int flag=1;
   for(int factor=2;factor<=sqrt(num);factor++)
   {
           if(num%factor==0)
   flag=0;
   }

  return flag;
}

void main()
{
  int Upper ,Lower;
  cout<<"Enter Lower Limit"<<endl;
  cin>>Lower;
  cout<<"Enter Upper limit"<<endl;
  cin>>Upper;
  if(Upper>=Lower)
  {cout<<"The Prime numbers between "<<Lower<<" & "<<Upper<<"are"<<endl;
   for(int i=Lower;i<=Upper;i++)
   {
           if(isprime(i))
           cout<<i<<"\t";
   }
  }
  else
  cout<<"Range Undefined";
  getch();
}
```



```
Enter Lower Limit
10
Enter Upper limit
100
The Prime numbers between 10 & 100are
11      13      17      19      23      29      31      37      41      43
47      53      59      61      67      71      73      79      83      89
97      ▬
```

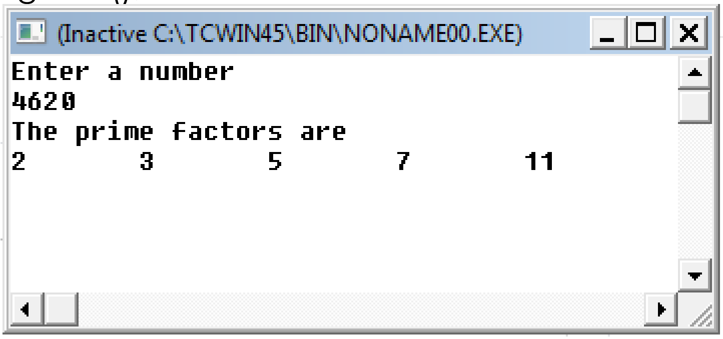
```
//3
/* A program to read the previous reading and the current reading and then
calculate and display monthly Telephone bill. The telephone bill is to be
calculated as per the following criterion:
      Minimum Charges      Rs. 250 as Rent and 60 calls free.
                         Plus Rs. 1.20 per call for the next 100 calls
                         Plus Rs. 1.00 per call for the next 100 calls
                         Plus Rs. 0.75 per call beyond 260 calls.*/

#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
 int prevR,curR,calls;
  float charge=250;
  cout<<"Enter previous reading"<<endl;
  cin>>prevR;
  cout<<"Enter current Reading"<<endl;
  cin>>curR;
  calls=curR-prevR;
  calls-=60;
  if(calls<=100)
  {
       charge+=calls*1.20;
       cout<<"Telephone bill amount is rupees "<<charge;
  }
  else
  {
       charge+=120;
   if(calls<=200)
   {
     calls-=100;
       charge+=calls;
     cout<<"Telephone bill amount is rupees "<<charge;
   }
   else
   {
       charge+=100;
     calls-=200;
     charge+=calls*0.75;
     cout<<"Telephone bill amount is rupees "<<charge;
   }
  }
  getch();
}
```

```
//4
/* A program to read previous reading and the current reading and then calculate
and display the monthly electricity bill. The bill is to be calculated as per the
following criterion:
          Minimum Charges          Rs. 150 for first 100 units
                                   Plus Rs. 1.00 per unit for the next 100 units
                                   Plus Rs. 0.75 per unit for the next 100 units
                                   Plus Rs. 0.50 per unit beyond 300 units.      */

#include<iostream.h>
#include<conio.h>
#include<math.h>
void main()
{
   int prevR,curR,units;
  float charge=150;
  cout<<"Enter previous reading"<<endl;
  cin>>prevR;
  cout<<"Enter current Reading"<<endl;
  cin>>curR;
  units=curR-prevR;
  if (units<=100)
  cout<<"Electricity bill amount is rupees "<<charge;

  else if(units<=200)
  {
    units-=100;
   charge+=units;
   cout<<"Electricity bill amount is rupees "<<charge;
  }
  else
  {
   charge+=120;
    if(units<=300)
    {
      units-=200;
          charge+=units*0.75;
      cout<<"Electricity bill amount is rupees "<<charge;
    }
    else
    {
          charge+=100;
      units-=300;
      charge+=units*0.5;
      cout<<"Electricity bill amount is rupees "<<charge;
    }
  }
  getch();
}
```



```
Enter previous reading
104
Enter current Reading
500
Electricity bill amount is rupees 418
```

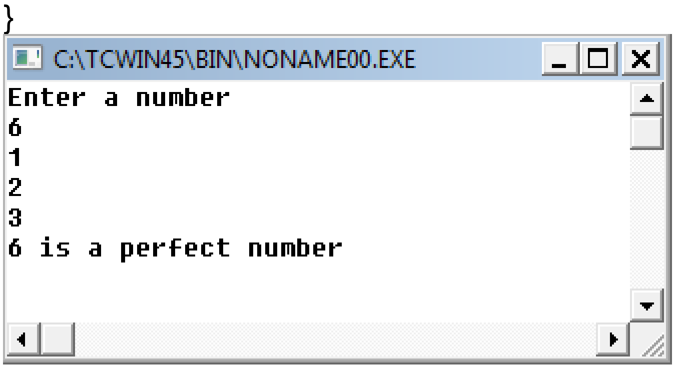
```
//5
//Display all prime factors of a number
#include<iostream.h>
#include<conio.h>
#include<math.h>
int isprime(int num)
{
  int flag=1;
  for(int factor=2;factor<=sqrt(num);factor++)
  {
          if(num%factor==0)
          flag=0;
  }

   return flag;
}
void main()
{
  int num,fac=2;
   cout<<"Enter a number"<<endl;
   cin>>num;
   if(isprime(num))
   cout<<"There are no prime factors";
   else
   cout<<"The prime factors are"<<endl;
   for(fac=2;fac<=num/2;fac++)
   {
          if(isprime(fac))
     {
          if(num%fac==0)
        cout<<fac<<"\t";
     }
   }
   getch();
```



```
Enter a number
4620
The prime factors are
2       3       5       7       11
```

```
}
```

```
//6
// To check whether a number is perfect or not
#include<iostream.h>
#include<conio.h>
void main()
{
   int num,fac[50];
  int i,j=0,count=0,sum=0;
  cout<<"Enter a number"<<endl;
  cin>>num;
  for(i=1;i<=num/2;i++)
  {
      if(num%i==0)
      fac[j++]=i;
      count++;

  }
  for(int k=0;k<count;k++)
  {
   cout<<fac[k]<<endl;
   sum+=fac[k];

  }
  cout<<sum;
  if(sum==num)
  cout<<"Its is a perfect number"<<endl;
  else
  cout<<"It is not a perfect number";

  getch();
}
```



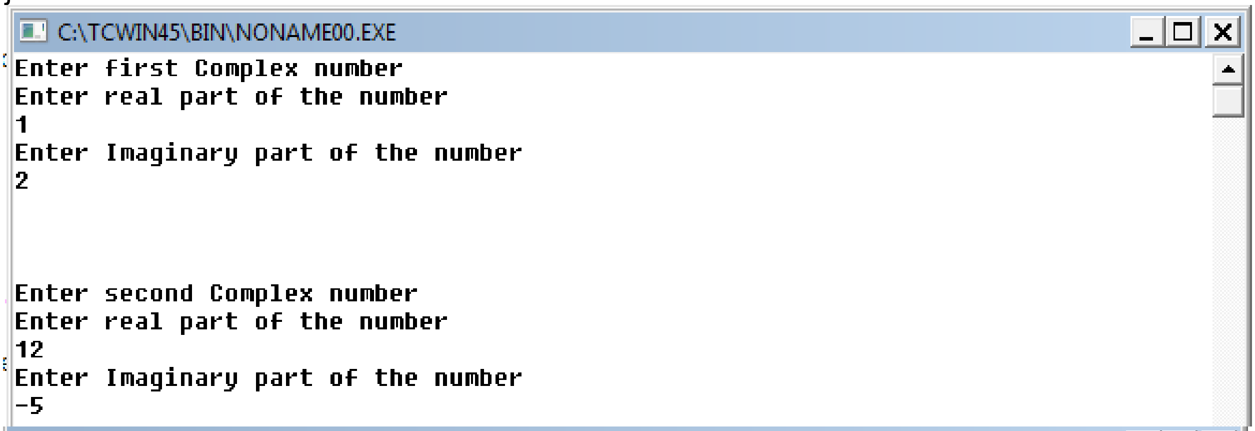
```
Enter a number
6
1
2
3
6 is a perfect number
```

//7
/*Menu based OOP to process the COMPLEX algebra (ADD, DIFFERENCE, MULTIPLICATION).*/

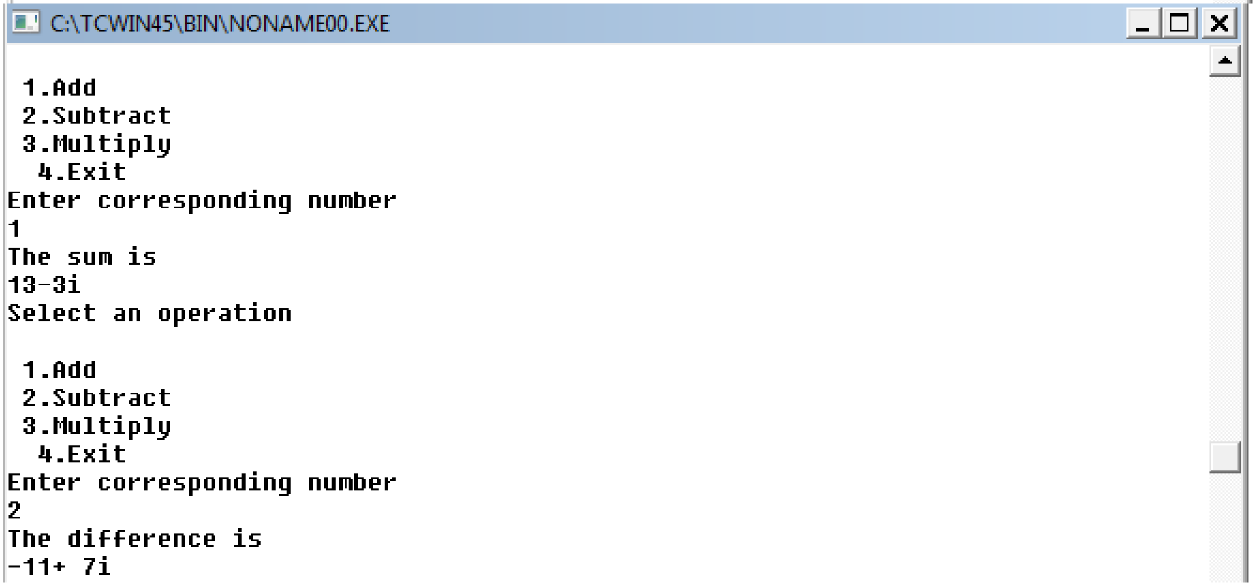
```
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>

class COMPLEX
{
        float im,rl;

  public:
        void getcomplex()
  {
        cout<<"Enter real part of the number"<<endl;
        cin>>rl;
        cout<<"Enter Imaginary part of the number"<<endl;
        cin>>im;
  }
  void showcomplex()
  {
        cout<<rl<<" ";
        if(im>0)
        cout<<"+ "<<im<<"i";
        else if(im<0)
        cout<<im<<"i";
  }
  void addcomplex(COMPLEX X,COMPLEX Y)
  {
        rl=X.rl+Y.rl;
        im=X.im+Y.im;
        cout<<"The sum is"<<endl<<rl;
        if(im>0)
        cout<<"+ "<<im<<"i";
        else if(im<0)
        cout<<im<<"i";

  }
  void diffcomplex(COMPLEX X,COMPLEX Y)
  {
        rl=X.rl-Y.rl;
        im=X.im-Y.im;
        cout<<"The difference is"<<endl<<rl;
        if(im>0)
        cout<<"+ "<<im<<"i";
        else if(im<0)
        cout<<im<<"i";

  }
        void mulcomplex(COMPLEX X,COMPLEX Y)
  {
        rl=X.rl*Y.rl;
```

```
       if(X.im*Y.im>0)
       rl-=X.im*Y.im;
       if(X.im*Y.im<0)
       rl+=X.im*Y.im;
       im=X.rl*Y.im+X.im*Y.rl;
       cout<<"The product is"<<endl<<rl;
       if(im>0)
       cout<<"+ "<<im<<"i";
       else if(im<0)
       cout<<im<<"i";
  }
};

void main()
{
  char op;
  COMPLEX C1,C2,C3;
  cout<<"Enter first Complex number"<<endl;
  C1.getcomplex();
  cout<<endl<<endl<<endl;
  cout<<"Enter second Complex number"<<endl;
  C2.getcomplex();
  cout<<endl<<endl<<endl;

  do
  {
 char op;
  cout<<"Select an operation"<<endl<<"\n 1.Add \n 2.Subtract \n”;
   cout<<” 3.Multiply \n  4.Exit"<<endl;
       cout<<"Enter corresponding number"<<endl;
       cin>>op;
     switch(op)
       {
            case '1':  C3.addcomplex(C1,C2);
                       break;
            case '2':  C3.diffcomplex(C1,C2);
                       break;
            case '3':  C3.mulcomplex(C1,C2);
                       break;
            case '4':   exit(0);
                       break;
            default :  cout<<"Invalid Input";
                       break;
       }
  }
  while(op!=4);
 getch();
```

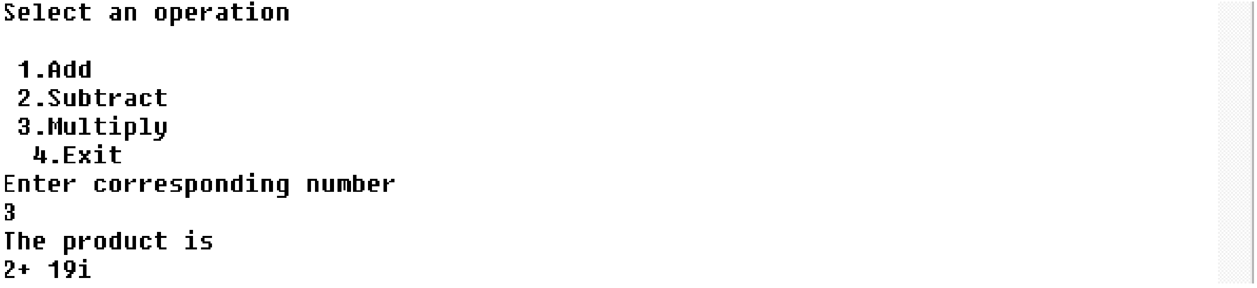
```
}
```



```
Enter first Complex number
Enter real part of the number
1
Enter Imaginary part of the number
2



Enter second Complex number
Enter real part of the number
12
Enter Imaginary part of the number
-5
```



```

  1.Add
  2.Subtract
  3.Multiply
    4.Exit
Enter corresponding number
1
The sum is
13-3i
Select an operation

  1.Add
  2.Subtract
  3.Multiply
    4.Exit
Enter corresponding number
2
The difference is
-11+ 7i
```



```
Select an operation

  1.Add
  2.Subtract
  3.Multiply
    4.Exit
Enter corresponding number
3
The product is
2+ 19i
```

//8

// Rewrite the above programs (problem 23) using ***structure***

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

struct COMPLEX
{
  private:
 float im,rl;
 public:
 void getcomplex()
 {
        cout<<"Enter real part of the number"<<endl;
        cin>>rl;
        cout<<"Enter Imaginary part of the number"<<endl;
        cin>>im;
```

```
    }
    void showcomplex()
    {
            cout<<rl<<" ";
            if(im>0)
            cout<<"+ "<<im<<"i";
            else if(im<0)
            cout<<im<<"i";
    }
    void addcomplex(COMPLEX X,COMPLEX Y)
    {
            rl=X.rl+Y.rl;
            im=X.im+Y.im;
            cout<<"The sum is"<<endl<<rl;
            if(im>0)
            cout<<"+ "<<im<<"i";
            else if(im<0)
            cout<<im<<"i";

    }
    void diffcomplex(COMPLEX X,COMPLEX Y)
    {
            rl=X.rl-Y.rl;
            im=X.im-Y.im;
            cout<<"The difference is"<<endl<<rl;
            if(im>0)
            cout<<"+ "<<im<<"i";
            else if(im<0)
            cout<<im<<"i";

    }

    void mulcomplex(COMPLEX X,COMPLEX Y)
    {
            rl=X.rl*Y.rl;
            if(X.im*Y.im>0)
            rl-=X.im*Y.im;
            if(X.im*Y.im<0)
            rl+=X.im*Y.im;
            im=X.rl*Y.im+X.im*Y.rl;
            cout<<"The product is"<<endl<<rl;
            if(im>0)
            cout<<"+ "<<im<<"i";
            else if(im<0)
            cout<<im<<"i";
    }
};

void main()
{
    char op;
```

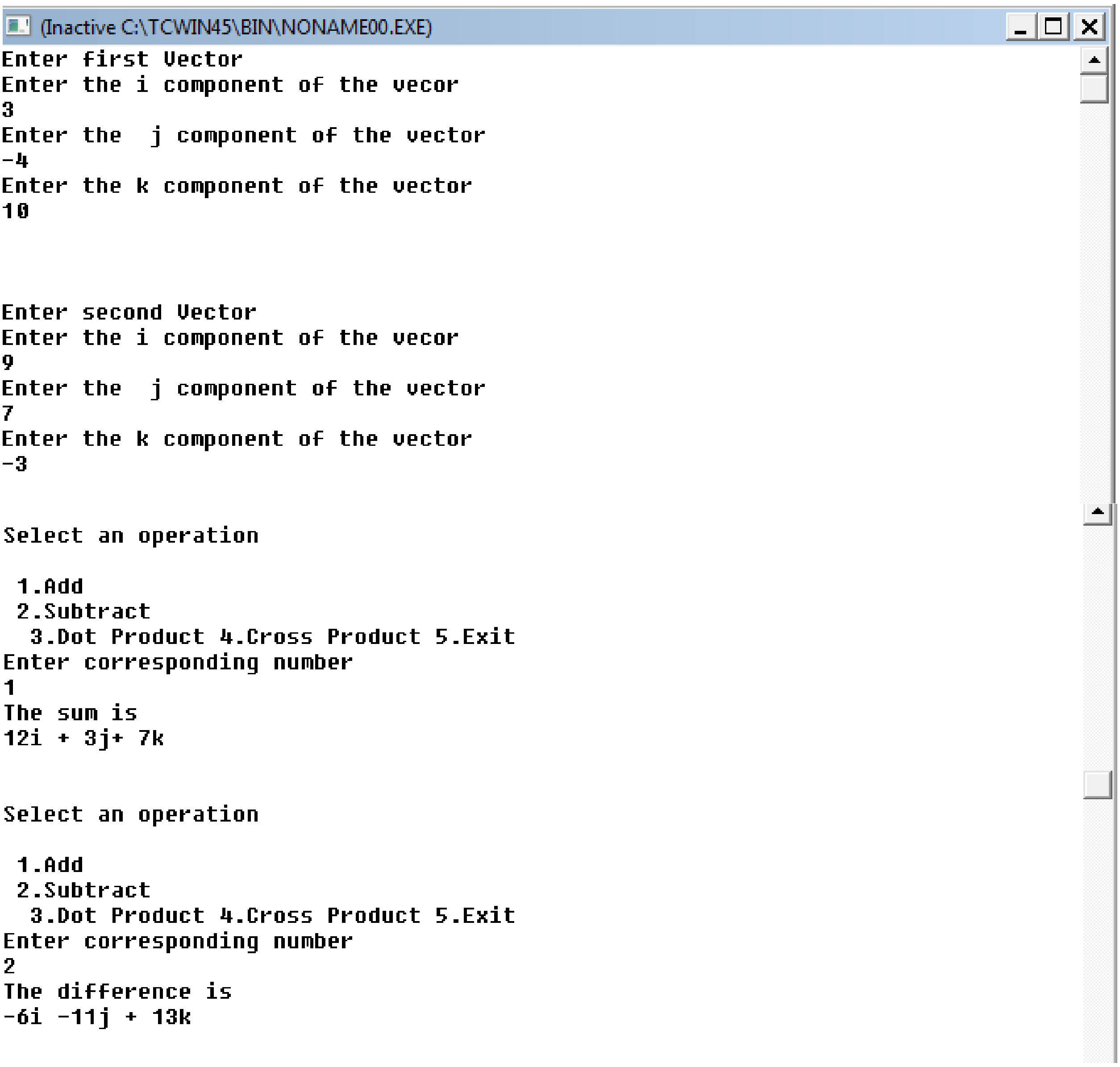
```
  COMPLEX C1,C2,C3;
  cout<<"Enter first Complex number"<<endl;
  C1.getcomplex();
  cout<<endl<<endl<<endl;
  cout<<"Enter second Complex number"<<endl;
  C2.getcomplex();
  cout<<endl<<endl<<endl;

  do
  {
          char op;
          cout<<"Select an operation"<<endl<<"\n 1.Add \n 2.Subtract \n”;
          cout<<” 3.Multiply \n  4.Exit"<<endl;
          cout<<"Enter corresponding number"<<endl;
          cin>>op;
          switch(op)
          {
                  case '1':       C3.addcomplex(C1,C2);
                                  break;
                  case '2':       C3.diffcomplex(C1,C2);
                                  break;
                  case '3':       C3.mulcomplex(C1,C2);
                                  break;
                  case '4':        exit(0);
                                  break;
                  default :       cout<<"Invalid Input";
                                  break;
          }
  }
  while(op!=4);
   getch();
}
```

//9

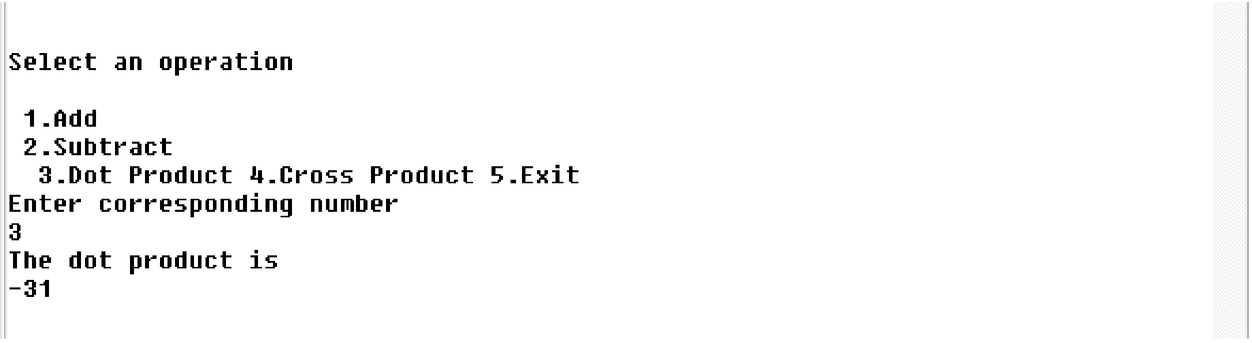
/*A menu based OOP to process the VECTOR algebra (ADD, DIFFERENCE, DOT PRODUCT and CROSS PRODUCT).*/

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

class VECTOR
{
  private:
   float j,i,k;

   public:
   void getvector()
   {
            cout<<"Enter the i component of the vecor"<<endl;
            cin>>i;
            cout<<"Enter the  j component of the vector"<<endl;
            cin>>j;
            cout<<"Enter the k component of the
vector"<<endl;
            cin>>k;
   }
   void showvector()
   {
            cout<<i<<"i ";
            if(j>0)
            cout<<"+ "<<j<<"j";
            else if(j<0)
            cout<<j<<"j ";
             if(k>0)
            cout<<"+ "<<k<<"k";
            else if(k<0)
            cout<<k<<"k";
   }
   void addvector(VECTOR X,VECTOR Y)
   {
            i=X.i+Y.i;
            j=X.j+Y.j;
             k=X.k+Y.k;
            cout<<"The sum is"<<endl<<i<<"i ";
            if(j>0)
            cout<<"+ "<<j<<"j";
            else if(j<0)
            cout<<j<<"j ";
             if(k>0)
            cout<<"+ "<<k<<"k";
            else if(k<0)
            cout<<k<<"k";
```

```
     }
     void diffvector(VECTOR X,VECTOR Y)
     {
          i=X.i-Y.i;
          j=X.j-Y.j;
          k=X.k-Y.k;
          cout<<"The difference is"<<endl<<i<<"i ";
          if(j>0)
          cout<<"+ "<<j<<"j";
          else if(j<0)
          cout<<j<<"j ";
          if(k>0)
          cout<<"+ "<<k<<"k";
          else if(k<0)
          cout<<k<<"k";
     }

     void dotproduct(VECTOR X,VECTOR Y)
     {
          i=X.i*Y.i;
          j=X.j*Y.j;
          k=X.k*Y.k;
          cout<<"The dot product is"<<endl<<i+j+k;
     }

      void crossproduct(VECTOR X,VECTOR Y)
     {
          i=(X.j*Y.k)-(X.k*Y.j);
          j=-1*((X.i*Y.k)-(X.k*Y.i));
          k=(X.i*Y.j)-(X.j*Y.i);
          cout<<"The cross product is"<<endl<<i+j+k;
     }
    };
void main()
    {
     char op;
     VECTOR V1,V2,V3;
     cout<<"Enter first Vector"<<endl;
     V1.getvector();
     cout<<endl<<endl<<endl;
     cout<<"Enter second Vector"<<endl;
     V2.getvector();
     cout<<endl<<endl<<endl;
    do
     {
          char op;
          cout<<endl<<endl<<endl;
          cout<<"Select an operation"<<endl<<"\n 1.Add \n 2.Subtract \n”;
          cout<<” 3.Dot Product 4.Cross Product 5.Exit"<<endl;
          cout<<"Enter corresponding number"<<endl;
          cin>>op;
          switch(op)
          {
                case '1':    V3.addvector(V1,V2);
                             break;
                case '2':    V3.diffvector(V1,V2);
                             break;
                case '3':    V3.dotproduct(V1,V2);
                             break;
                case '4':    V3.crossproduct(V1,V2);
                             break;
          case '5':    exit(0);
                       break;
          default :    cout<<"Invalid Input";
                       break;
     }
    }
  while(op!=5);
  getch();
  }
```



```
Enter first Vector
Enter the i component of the vecor
3
Enter the  j component of the vector
-4
Enter the k component of the vector
10



Enter second Vector
Enter the i component of the vecor
9
Enter the  j component of the vector
7
Enter the k component of the vector
-3


Select an operation

 1.Add
 2.Subtract
  3.Dot Product 4.Cross Product 5.Exit
Enter corresponding number
1
The sum is
12i + 3j+ 7k


Select an operation

 1.Add
 2.Subtract
  3.Dot Product 4.Cross Product 5.Exit
Enter corresponding number
2
The difference is
-6i -11j + 13k
```



```
Select an operation

 1.Add
 2.Subtract
  3.Dot Product 4.Cross Product 5.Exit
Enter corresponding number
3
The dot product is
-31
```

//10

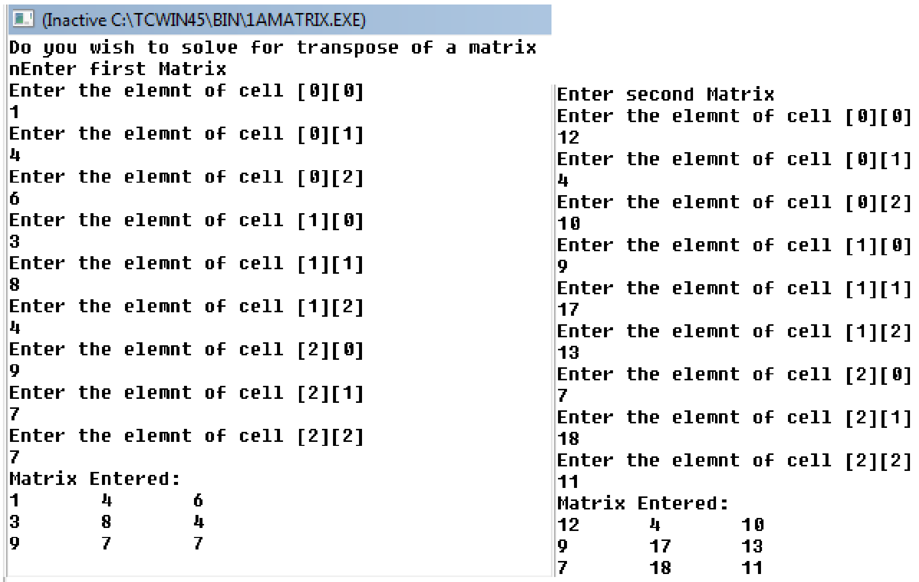
//Rewrite the above problem (problem 25)using structure

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

struct VECTOR
{
   private:

  float j,i,k;

  public:
          void getvector()
  {
          cout<<"Enter the i component of the vecor"<<endl;
          cin>>i;
          cout<<"Enter the  j component of the vector"<<endl;
          cin>>j;
                                                  cout<<"Enter the k component of the
                          vector"<<endl;
          cin>>k;
  }
  void showvector()
  {
          cout<<i<<"i ";
          if(j>0)
          cout<<"+ "<<j<<"j";
          else if(j<0)
          cout<<j<<"j ";
           if(k>0)
          cout<<"+ "<<k<<"k";
          else if(k<0)
          cout<<k<<"k";
  }
  void addvector(VECTOR X,VECTOR Y)
  {
          i=X.i+Y.i;
          j=X.j+Y.j;
           k=X.k+Y.k;
          cout<<"The sum is"<<endl<<i<<"i ";
          if(j>0)
          cout<<"+ "<<j<<"j";
```

```
            else if(j<0)
            cout<<j<<"j ";
             if(k>0)
            cout<<"+ "<<k<<"k";
            else if(k<0)
            cout<<k<<"k";

    }
    void diffvector(VECTOR X,VECTOR Y)
    {
            i=X.i-Y.i;
            j=X.j-Y.j;
            k=X.k-Y.k;
            cout<<"The difference is"<<endl<<i<<"i ";
            if(j>0)
            cout<<"+ "<<j<<"j";
            else if(j<0)
            cout<<j<<"j ";
            if(k>0)
            cout<<"+ "<<k<<"k";
            else if(k<0)
            cout<<k<<"k";

    }

    void dotproduct(VECTOR X,VECTOR Y)
    {
            i=X.i*Y.i;
            j=X.j*Y.j;
            k=X.k*Y.k;
            cout<<"The dot product is"<<endl<<i+j+k;
    }

   void crossproduct(VECTOR X,VECTOR Y)
    {
            i=(X.j*Y.k)-(X.k*Y.j);
            j=-1*((X.i*Y.k)-(X.k*Y.i));
            k=(X.i*Y.j)-(X.j*Y.i);
            cout<<"The cross product is"<<endl<<i+j+k;
    }
};
void main()
{
    char op;
    VECTOR V1,V2,V3;
    cout<<"Enter first Vector"<<endl;
    V1.getvector();
    cout<<endl<<endl<<endl;
    cout<<"Enter second Vector"<<endl;
    V2.getvector();
    cout<<endl<<endl<<endl;
    do
     {
          char op;
          cout<<endl<<endl<<endl;
          cout<<"Select an operation"<<endl<<"\n 1.Add \n 2.Subtract \n”;
          cout<<” 3.Dot Product 4.Cross Product 5.Exit"<<endl;
          cout<<"Enter corresponding number"<<endl;
          cin>>op;
          switch(op)
          {
               case '1':     V3.addvector(V1,V2);
                             break;
               case '2':     V3.diffvector(V1,V2);
                             break;
               case '3':     V3.dotproduct(V1,V2);
                             break;
               case '4':     V3.crossproduct(V1,V2);
                             break;
          case '5':     exit(0);
                        break;
          default :     cout<<"Invalid Input";
                        break;
     }
     }
    while(op!=5);
    getch();
}
```

```
/*11
A menu based OOP to process the MATRIX algebra (ADD, DIFFERENCE,
MULTIPLICATION and TRANSPOSE).
*/
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>

const SIZE=3;

class MATRIX
{


        float mat[SIZE][SIZE];

    public:

        void getmatrix()
        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    cout<<"Enter the elemnt of cell ["<<i<<"]["<<j<<"] \n";
                    cin>>mat[i][j];
                }
            }
            cout<<"Matrix Entered: \n";
        showmatrix();
        }

        void showmatrix()
        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    cout<<mat[i][j]<<'\t';
                }
                    cout<<endl;
            }
        }


        void addmatrices(MATRIX X,MATRIX Y)
        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    mat[i][j]=X.mat[i][j]+Y.mat[i][j];
                }
            }

            cout<<"The sum is"<<endl;
            showmatrix();

        }
        void diffmatrices(MATRIX X,MATRIX Y)
        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    mat[i][j]=X.mat[i][j]-Y.mat[i][j];
                }
            }
            cout<<"The difference is"<<endl;
            showmatrix();

        }

         void mulmatrices(MATRIX X, MATRIX Y)
//c[i][j]=a[i][0].b[0][j]+a[i][1].b[1][j]+…+a[i][n].b[n][j]

        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    float cellsum=0;
                    for(int n=0;n<SIZE;n++)
                    {
                        cellsum+=X.mat[i][n]*Y.mat[n][j];
                    }
                    mat[i][j]=cellsum;
                }
            }

            cout<<"The product is \n";
            showmatrix();
        }

        void transposematrix(MATRIX X)
        {
            for(int i=0;i<SIZE;i++)
            {
                for(int j=0;j<SIZE;j++)
                {
                    mat[i][j]=X.mat[j][i];
            }
```

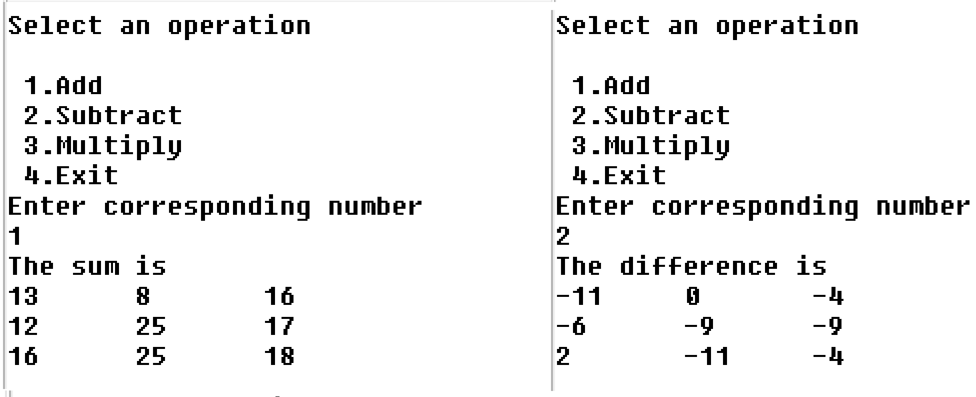
```
                        }
    cout<<"\n\n Transpose of the matrix is:\n";
                        showmatrix();
                }

    };



    void main()
    {
       char op,typ;
       cout<<"Do you wish to solve for transpose of a matrix \n";
       typ=getche();


       if(typ=='N' || typ=='n')
                goto binaryoperation;  //send to level

       else if(typ=='Y' || typ=='y')
                cout<<"Enter Matrix \n";
       MATRIX M;
       M.getmatrix();
       M.transposematrix(M);
       exit(0);


       binaryoperation:                          //level name
       MATRIX M1,M2,M3;
       cout<<"Enter first Matrix"<<endl;
       M1.getmatrix();
       cout<<endl<<endl<<endl;
       cout<<"Enter second Matrix"<<endl;
       M2.getmatrix();
       do
       {
       char op;
       cout<<endl<<endl<<endl;
       cout<<"Select an operation"<<endl<<" \n 1.Add \n 2.Subtract \n 3.Multiply
    \n 4.Exit"<<endl;
       cout<<"Enter corresponding number"<<endl;
       cin>>op;

       switch(op)
                        {
                                case '1':       M3.addmatrices(M1,M2);
                                                break;
                                case '2':       M3.diffmatrices(M1,M2);
                                                break;
                                case '3':       M3.mulmatrices(M1,M2);
                                                break;
                        case '4':       exit(1);
                                        break;
                        default :       cout<<"Invalid Input";
                }
  }
  while(op!='4');
  getch();
}
```



(Inactive C:\TCWIN45\BIN\1AMATRIX.EXE)

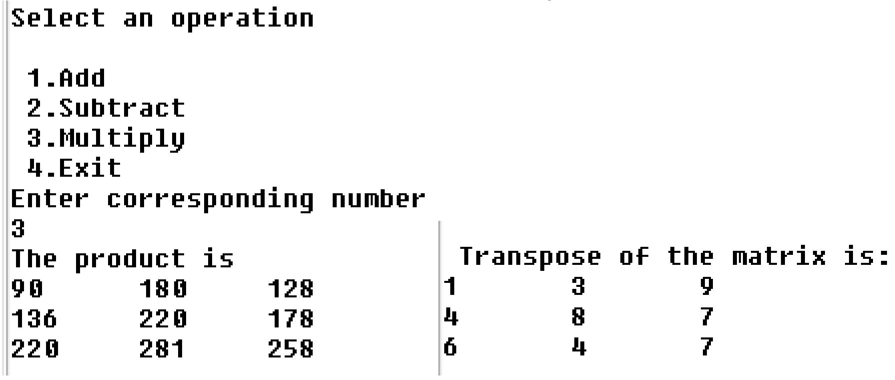
```
Do you wish to solve for transpose of a matrix
nEnter first Matrix
Enter the elemnt of cell [0][0]
1
Enter the elemnt of cell [0][1]
4
Enter the elemnt of cell [0][2]
6
Enter the elemnt of cell [1][0]
3
Enter the elemnt of cell [1][1]
8
Enter the elemnt of cell [1][2]
4
Enter the elemnt of cell [2][0]
9
Enter the elemnt of cell [2][1]
7
Enter the elemnt of cell [2][2]
7
Matrix Entered:
1       4       6
3       8       4
9       7       7
```

```
Enter second Matrix
Enter the elemnt of cell [0][0]
12
Enter the elemnt of cell [0][1]
4
Enter the elemnt of cell [0][2]
10
Enter the elemnt of cell [1][0]
9
Enter the elemnt of cell [1][1]
17
Enter the elemnt of cell [1][2]
13
Enter the elemnt of cell [2][0]
7
Enter the elemnt of cell [2][1]
18
Enter the elemnt of cell [2][2]
11
Matrix Entered:
12      4       10
9       17      13
7       18      11
```



```
Select an operation

 1.Add
 2.Subtract
 3.Multiply
 4.Exit
Enter corresponding number
1
The sum is
13      8       16
12      25      17
16      25      18
```

```
Select an operation

 1.Add
 2.Subtract
 3.Multiply
 4.Exit
Enter corresponding number
2
The difference is
-11     0       -4
-6      -9      -9
2       -11     -4
```



```
Select an operation

 1.Add
 2.Subtract
 3.Multiply
 4.Exit
Enter corresponding number
3
The product is
90      180     128
136     220     178
220     281     258
```

```
 Transpose of the matrix is:
1       3       9
4       8       7
6       4       7
```

```
//12
/*Declare and define a class EMPLOYEE having the following members. Also
write a menu based program to implement this class for 5 employee.

private : name, basic
da( )   // returns DA, which is 60% of the BASIC.
public :  getdata( )
showdata( )
*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class Employee
{
        char name[20];
  float basic;

  float da()
  {
             float DA;
        DA=basic*0.6;
             return DA;
  }

 public:

 void getdata()
 {
  cout<<"Enter Name"<<endl;
        gets(name);
        cout<<"Enter Basic salary"<<endl;
        cin>>basic;
 }
 void showdata()
 {
  cout<<"Name:-  "<<name<<endl<<"Basic :-  "<<basic<<endl;
        cout<<"DA :-  "<<da();
 }
};

void main()
{
  Employee E[5];
 int i;
 cout<<"Enter Employee number"<<endl;
 cin>>i;
 E[i].getdata();
 E[i].showdata();
 getch();
}
```

```
//13
/*Declare and define a class BANK having the following members. Also write a menu
based program to implement this class for 5 Account holders.
private:        name, balance, withdraw( ), deposit( )
public:getdata( ),showdata( ),transaction( )// to invoke withdraw( ) and
      deposit( ) depending upon user’s choice
*/
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
class BANK
{
  char name[30];
  float balance;
  void withdraw()
  {
    float WDamt;
    cout<<"Enter Amount to be withdrawn"<<endl;
    cin>>WDamt;
    if(WDamt>balance)
    cout<<"Insufficient funds"<<endl;
    else
    {
               balance-=WDamt;
               cout<<"Withdrawal successful"<<endl<<"Updated Account”;
               cout<<” balance is “<<balance<<endl;
    }
 }
void deposit()
  {
    float DPamt;
    cout<<"Enter Amount to be deposited"<<endl;
    cin>>DPamt;
    balance+=DPamt;
    cout<<"Deposit successful"<<endl<<"Updated Account balance is”;
    cout<<balance<<endl;
  }
public:
void getdata()
  {
       cout<<"Enter Name"<<endl;
       cin>>name;
       cout<<"Enter Account balance"<<endl;
       cin>>balance;
      }
      void showdata()
      {
       cout<<"Name :-  "<<name<<endl<<"Balance :- "<<balance<<endl;;
      }
      void transaction(BANK X)
```

```
{
   char tr;
    do
  {cout<<"Select transaction"<<endl<<" \n 1.Deposit \n”;
   cout<<” 2.Withdraw  \n  3.Exit"<<endl;
           cout<<"Enter corresponding number"<<endl;
           cin>>tr;
         switch(tr)
           {        case '1':       X.deposit();
                                    break;
                    case '2':       X.withdraw();
                                    break;
                    case '3':               exit(1);
                                    break;
                    default :       cout<<"Invalid Input";
                                    break;
           }
      }while(tr!=3);
   }
};
void main()
{
   int i;
   char op;
   BANK B[5];
   do
   {
           cout<<"Enter Account number "<<endl;
           cin>>i;
           B[i].getdata();
           do
          {
                    cout<<"How can I help you today?"<<endl;
                    cout<<"\n 1. Display Account details \n 2.Make a”;
                    cout<<” transaction \n 3.Exit”<<endl;
                    cin>>op;
                    switch(op)
                    {
                             case '1':      B[i].showdata();
                                            break;
                             case '2':      B[i].transaction(B[i]);
                                            break;
                             case '3':      exit(0);
                                            break;
                             default:       cout<<”Please Enter Relevent value”<<endl;
                                            break;
                    }
            }while(op!=3);

   }while(i!=0);
   getch();
```

```
}

//14
/*Declare and define a class STUDENT having the following members. Also write a menu
based program to implement this class for 10 students.

        private:                name (string), marks[5] //(an array of integers)
        result( )               // returns RESULT, which is "PASS" if  Total>=cutoff else "FAIL"
        public:                         getdata( )
                                showdata( )

        NOTE: Total is the sum of all marks: marks[0]..marks[4]*/

#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
class STUDENT
{
        char name[20];
  float marks[5];

  int result()
  {
        int result=0;
        float Total=0;
        for(int i=0;i<5;i++)
        Total+=marks[i];
        float cutoff=165;
        if(Total>cutoff)
        result=1;
        return result;
  }

  public:

  void getdata()
  {
        cout<<"Enter Name of the student"<<endl;
        gets(name);
        for(int i=0;i<5;i++)
        {
                cout<<"Enter Marks in subject "<<i+1<<endl;
                cin>>marks[i];
        }
  }
```

```
void showdata()
  {
        float total;
        cout<<"Name:-  "<<name<<endl;
        for(int i=0;i<5;i++)
        {
                cout<<"Marks in subject "<<i+1<<":-  "<<marks[i]<<endl;
                total+=marks[i];
        }
        cout<<"Total Marks:-  "<<total<<endl;
        cout<<"Result :-  ";

        if(result())
        cout<<"PASS"<<endl<<endl<<endl;
        else
        cout<<"FAIL"<<endl<<endl<<endl;
  }

};

void main()
{
        STUDENT S[5];
        S[0].getdata();
        char op;
        int i=1;
        do
        {
cout<<"Select Action"<<endl<<"\n 1. New Student \n 2. View Existing”;
cout<<”Student  \n 3.Exit"<<endl;
                cin>>op;
                switch(op)
                {
                case '1':       S[i].getdata();
                                i++;
                                break;
                case '2':       int j;
                                cout<<"Enter Student Index"<<endl;
                                cin>>j;
                                S[j].showdata();
                                break;
                case '3':       exit(0);
                                break;
                default :       cout<<"Invalid Input"<<endl;
                                break;
                }
        }while (op!=3);
        getch();
   }
```

//15
/* Declare and define a structure TIME having the member variables hour, minutes. Using the concept of <u>Containership</u>, also declare and define a class TRAIN having member variables Train_no (integer type), Train_name (string type), arrival_time (TIME type) and departure_time (TIME type).

Write a simple (minimum) menu based program to process the above structure and class.*/

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
struct TIME
{
        int hours, minutes;
};

class TRAIN
{
        int Train_Number;
        char Train_Name[20];
        TIME arrival_time,departure_time;

   public:

        void getdata()
        {
                cout<<"Enter Train Number"<<endl;
                cin>>Train_Number;
                cout<<"Enter Train Name"<<endl;
                gets(Train_Name);
                cout<<"Enter Train Arrival Time"<<endl<<endl<<endl;
                cout<<"Enter Train Arrival Time Hour"<<endl;
                cin>>arrival_time.hours;
                cout<<"Enter Train Arrival Time Minutes"<<endl;
                cin>>arrival_time.minutes;
                cout<<"Enter Train departure Time"<<endl<<endl<<endl;
                cout<<"Enter Train departure Time Hour"<<endl;
                cin>>departure_time.hours;
                cout<<"Enter Train departure Time Minutes"<<endl;
                cin>>departure_time.minutes;
        }
```

```
    void showdata()
    {
        cout<<"Train Number:-  "<<Train_Number<<endl;
        cout<<"Train Name:-  "<<Train_Name<<endl;
        cout<<"Train Arrival Time:-  "<<arrival_time.hours<<"hours”;
    cout"<<arrival_time.minutes<<"minutes"<<endl;
cout<<"Train departure Time:-  "<<departure_time.hours<<"hours”;
cout<<departure_time.minutes<<"minutes"<<endl;
    }
};

void main()
{
    TRAIN T;
    cout<<"Enter train details"<<endl;
    T.getdata();
    char a;
    do
    {
    cout<<"1. View Train details \n 2.Exit \n”;
    cout<<” Select corresponding Number"<<endl;
        cin>>a;
        switch (a)
        {
            case'1':        T.showdata();
                            break;
            case'2':        exit(0);
                            break;
            default:        cout<<"Invalid input"<<endl;
                            break;
        }
    }while(a!=2);
  getch();
}
```

```
//16
/*Declare and define a class PERSON having the member variables name, address, phone with necessary member functions. Using the concept of Inheritance, also declare and define another class STUDENT having all the members of PERSON with some extra member variables – class, sec, roll and required necessary member functions.
Write a simple (minimum) menu based program to process 5 STUDENTS.
*/
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
class PERSON
{
   protected:

        char Name[20],address[50];
        long phone_no;

  public:

        void getdata()
        {
                cout<<"Enter Name"<<endl;
                gets(Name);
                cout<<"Enter Address"<<endl;
                gets(address);
                cout<<"Enter Phone number"<<endl;
                cin>>phone_no;
        }
        void showdata()
        {
                cout<<"Name:-  "<<Name<<endl;
                cout<<"Address:-  "<<address<<endl;
                cout<<"Phone No."<<phone_no<<endl;
        }
};


class STUDENT:private PERSON
{
        char class_[3],section;
        int roll_number;
        public:
        void getinfo()
        {
                getdata();
                cout<<"Enter Class"<<endl;
                gets(class_);
                cout<<"Enter Section"<<endl;
                cin>>section;
                cout<<"Enter Roll Number"<<endl;
                cin>>roll_number;
        }
        void showinfo()
        {
                showdata();
                cout<<"Class & Section:-  "<<class_<<section<<endl;
                cout<<"Roll Number:-  "<<roll_number<<endl;
        }
};


void main()
{
        STUDENT S[5];
        S[0].getinfo();
        char op;
        int i=1;
        do
        {
                cout<<"Select Action"<<endl<<"1. New Student \n”;
        cout<<” 2. View Existing Student \n 3.Exit"<<endl;
                cin>>op;
                switch(op)
                {
                        case '1':       S[i].getinfo();
                                        i++;
                                        break;

                        case '2':       int j;
                                        cout<<"Enter Student Index"<<endl;
                                        cin>>j;
                                        S[j].showinfo();
                                        break;
                        case '3':       exit(0);
                                        break;
                        default :       cout<<"Invalid Input"<<endl;
                                        break;
                }
        }while (op!=3);
}
```

```
//17
/*Declare and define three functions having same name area( ) to calculate area of
square, rectangle and triangle (using HERON's formula). Write the complete menu based
to use these functions effectively.*/
#include<iostream.h>
#include<conio.h>
#include<math.h>
float area(float S)
{
        return S*S;
}
float area(float A,float B)
{
        return A*B;
}
float area(float A,float B,float C)
{
        float area,S;
        S=(A+B+C)/2;
        area=sqrt(S*(S-A)*(S-B)*(S-C));
        return area;
}
void main()
{
        char sh;
        cout<<"Select shape"<<endl;
        cout<<"1. Square \n 2. Rectangle \n 3. Triangle"<<endl;
        cin>>sh;
        switch(sh)
        {
                case '1':       float side;
                                cout<<"Enter side"<<endl;
                                cin>>side;
                                cout<<"Area of square "<<area(side)<<"sq units";
                                break;
                case '2':       float Length,Breadth;
                                cout<<"Enter Length"<<endl;
                                cin>>Length;
                                cout<<"Enter Breadth"<<endl;
                                cin>>Breadth;
                                cout<<"Area of rectangle "<<area(Length,Breadth)<<"sq units";
                                break;
                case '3':       float A,B,C;
                                cout<<"Enter side A"<<endl;
                                cin>>A;
                                cout<<"Enter side B"<<endl;
                                cin>>B;
                                cout<<"Enter side C"<<endl;
                                cin>>C;
                                if(area(A,B,C)==0)
                                cout<<"Invalid Triangle";
                                else
                                cout<<"Area of the triangle is "<<area(A,B,C)<<"sq units";
                                break;

                default:                cout<<"Invalid input";

        }
  getch();
}
```

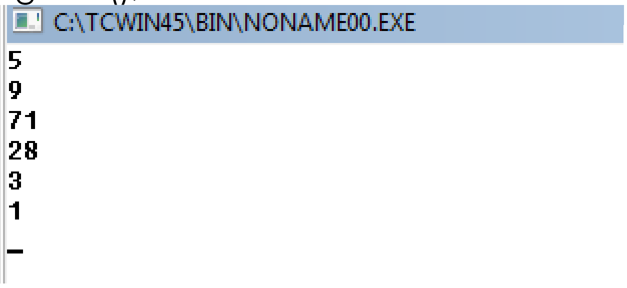
//18

/*Define the following functions having same name DICE( ) and also write the required program to demonstrate their use.

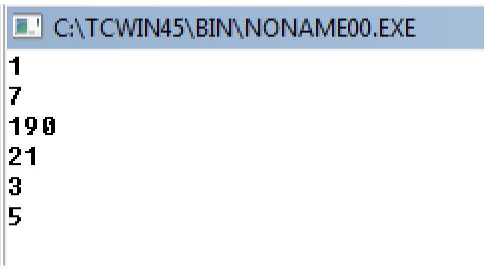
A If one argument N (having default value 6) is passed, the function should return a random value between 1 and N (both inclusive).

B If two arguments M & N are passed, then the function should return a random value between M and N (both inclusive).*/

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

int dice(int n=6)
{
        return 1+random(n);
}
int dice(int n,int m)
{
        return m+random(n+1);
}
void main()
{
  randomize();
  cout<<dice()<<endl;
  cout<<dice(10)<<endl;
  cout<<dice(200)<<endl;
  cout<<dice(10,20)<<endl;
  cout<<dice(0,3)<<endl;
        cout<<dice()<<endl;
  getch();
```

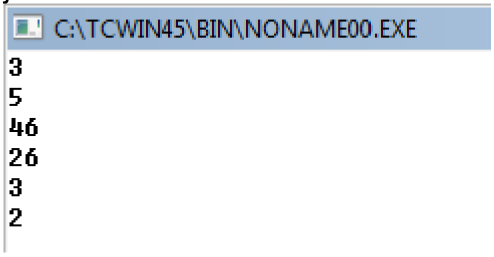


```
5
9
71
28
3
1
_
```



```
1
7
190
21
3
5
```

```
}
```

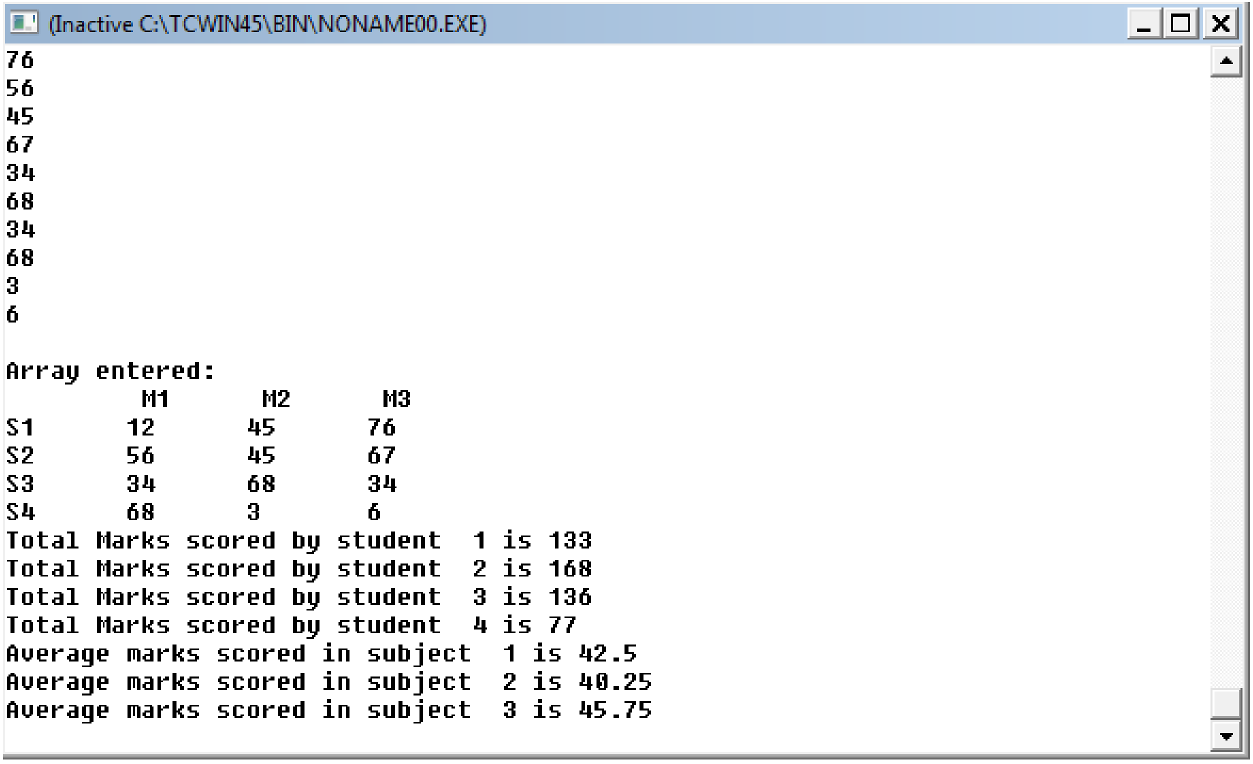


```
3
5
46
26
3
2
```

//19

/*WAP to declare and read the marks of 4 students in 3 subjects (use 2-D array) and then calculate their total. Also calculate subject wise average marks scored by the students.*/

```
#include<iostream.h>
#include<conio.h>
void main()
{
        cout<<"Enter 2-D array row wise"<<endl;
        float S[4][3];
        int i,j;
        for(i=0;i<4;i++)
        {
                for(j=0;j<3;j++)
                cin>>S[i][j];
        }
        cout<<endl<<"Array entered:"<<endl;
        cout<<"\t M1 \t M2 \t M3"<<endl;
        for(int k=0;k<4;k++)
        {
                cout<<"S"<<k+1<<'\t';
                for(int l=0;l<3;l++)
                {
                        cout<<S[k][l]<<"\t";
                }
                cout<<endl;
        }
        for(i=0;i<4;i++)
        {
                float marks=0;
                for(int j=0;j<3;j++)
                {
                        marks+=S[i][j];
                }
                cout<<"Total Marks scored by student  "<<i+1<<" is "<<marks<<endl;
        }
        for(j=0;j<3;j++)
        {
    float Subjtotal=0;
                for(i=0;i<4;i++)
                {
                        Subjtotal+=S[i][j];
                }
                cout<<"Average marks scored in subject  "<<j+1<<" is "<<Subjtotal/4<<endl;
        }

        getch();
}
```



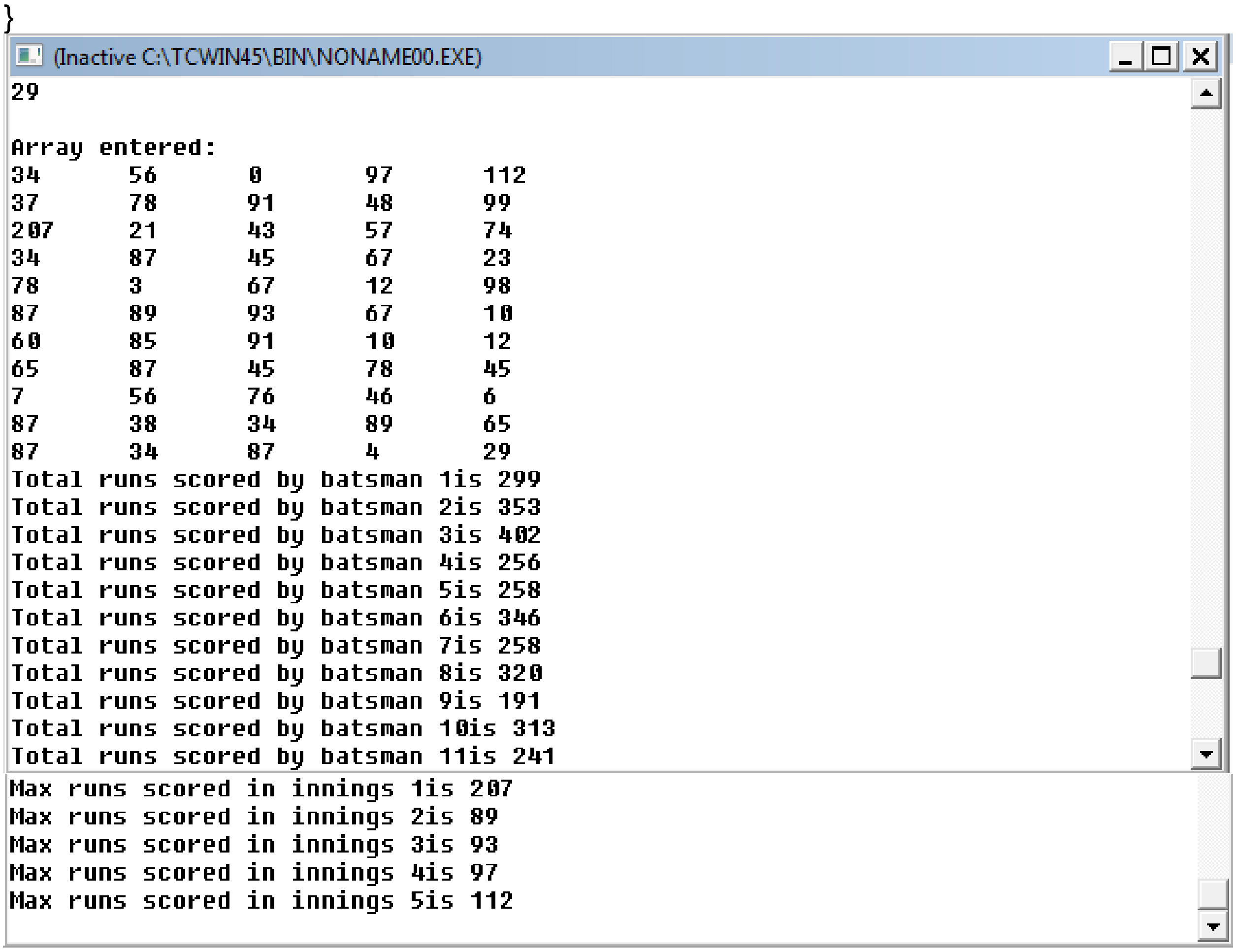
```
76
56
45
67
34
68
34
68
3
6

Array entered:
        M1      M2      M3
S1      12      45      76
S2      56      45      67
S3      34      68      34
S4      68      3       6
Total Marks scored by student  1 is 133
Total Marks scored by student  2 is 168
Total Marks scored by student  3 is 136
Total Marks scored by student  4 is 77
Average marks scored in subject  1 is 42.5
Average marks scored in subject  2 is 40.25
Average marks scored in subject  3 is 45.75
```

//20

/* WAP to accept runs scored by 11 batsmen in 5 innings in a 2d array. Also show the total runs scored by each batsman and the maximum runs scored in each innings*/

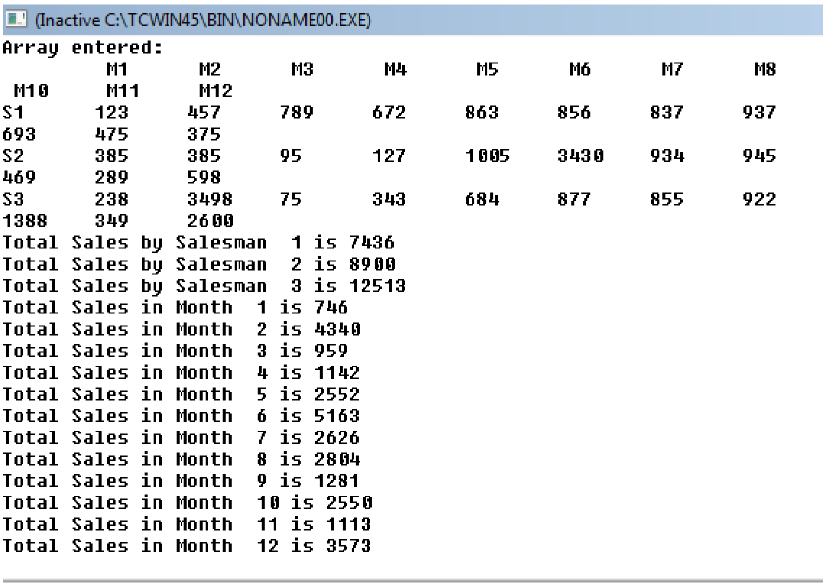
```
#include<iostream.h>
#include<conio.h>
void main()
{
	cout<<"Enter 2-D array row wise"<<endl;
	int S[11][5],i,j;
	for(i=0;i<11;i++)
	{
		for(j=0;j<5;j++)
		cin>>S[i][j];
	}
	cout<<endl<<"Array entered:"<<endl;
	for(int k=0;k<11;k++)
	{
		for(int l=0;l<5;l++)
		{
			cout<<S[k][l]<<"\t";
		}
		cout<<endl;
	}
	for(i=0;i<11;i++)
	{
		int runs=0;
		for(int j=0;j<5;j++)
		{
			runs+=S[i][j];
		}
		cout<<"Total runs scored by batsman "<<i+1<<"is "<<runs<<endl;
	}
```

```
        for(j=0;j<5;j++)
        {
                int max=S[0][j];
                for(i=0;i<11;i++)
                {
                        if(S[i][j]>max)
                        {
                                max=S[i][j];
                        }
                }

                cout<<"Max runs scored in innings "<<j+1<<"is "<<max<<endl;
        }
        getch();
}
```



```
29

Array entered:
34      56      0       97      112
37      78      91      48      99
207     21      43      57      74
34      87      45      67      23
78      3       67      12      98
87      89      93      67      10
60      85      91      10      12
65      87      45      78      45
7       56      76      46      6
87      38      34      89      65
87      34      87      4       29
Total runs scored by batsman 1is 299
Total runs scored by batsman 2is 353
Total runs scored by batsman 3is 402
Total runs scored by batsman 4is 256
Total runs scored by batsman 5is 258
Total runs scored by batsman 6is 346
Total runs scored by batsman 7is 258
Total runs scored by batsman 8is 320
Total runs scored by batsman 9is 191
Total runs scored by batsman 10is 313
Total runs scored by batsman 11is 241
Max runs scored in innings 1is 207
Max runs scored in innings 2is 89
Max runs scored in innings 3is 93
Max runs scored in innings 4is 97
Max runs scored in innings 5is 112
```

```
//21
/*WAP to read the sales (in Rs.) made by 3 Salesmen in 12 Months (use 2-D  array) and
then calculate their annual Sales (in Rs.). Also calculate Month wise total sales made by
the salesmen.*/
#include<iostream.h>
#include<conio.h>
void main()
{
        cout<<"Enter 2-D array row wise"<<endl;
        float S[3][12];
        int i,j;
        for(i=0;i<3;i++)
        {
                for(j=0;j<12;j++)
                cin>>S[i][j];
        }
        cout<<endl<<"Array entered:"<<endl;
        for(int z=1;z<13;z++)
        cout<<"\t M"<<z;
        cout<<endl;
        for(int k=0;k<3;k++)
        {
                cout<<"S"<<k+1<<'\t';
                for(int l=0;l<12;l++)
                {
                        cout<<S[k][l]<<"\t";
                }
                cout<<endl;
        }
        for(i=0;i<3;i++)
        {
                float saltotal=0;
                for(int j=0;j<12;j++)
                {
                        saltotal+=S[i][j];
                }
                cout<<"Total Sales by Salesman  "<<i+1<<" is "<<saltotal<<endl;
        }
        for(j=0;j<12;j++)
        {
    float SalesM=0;
                for(i=0;i<3;i++)
                {
                        SalesM+=S[i][j];
                }
                cout<<"Total Sales in Month  "<<j+1<<" is "<<SalesM<<endl;
        }
        getch();
}
```

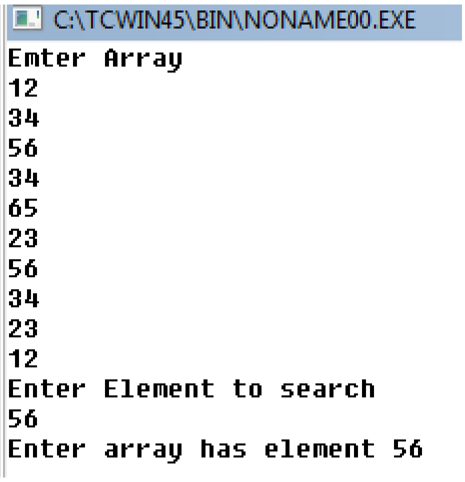


```
Array entered:
         M1       M2       M3       M4       M5       M6       M7       M8
 M10      M11      M12
S1       123      457      789      672      863      856      837      937
693      475      375
S2       385      385      95       127      1005     3430     934      945
469      289      598
S3       238      3498     75       343      684      877      855      922
1388     349      2600
Total Sales by Salesman  1 is 7436
Total Sales by Salesman  2 is 8900
Total Sales by Salesman  3 is 12513
Total Sales in Month  1 is 746
Total Sales in Month  2 is 4340
Total Sales in Month  3 is 959
Total Sales in Month  4 is 1142
Total Sales in Month  5 is 2552
Total Sales in Month  6 is 5163
Total Sales in Month  7 is 2626
Total Sales in Month  8 is 2804
Total Sales in Month  9 is 1281
Total Sales in Month  10 is 2550
Total Sales in Month  11 is 1113
Total Sales in Month  12 is 3573
```

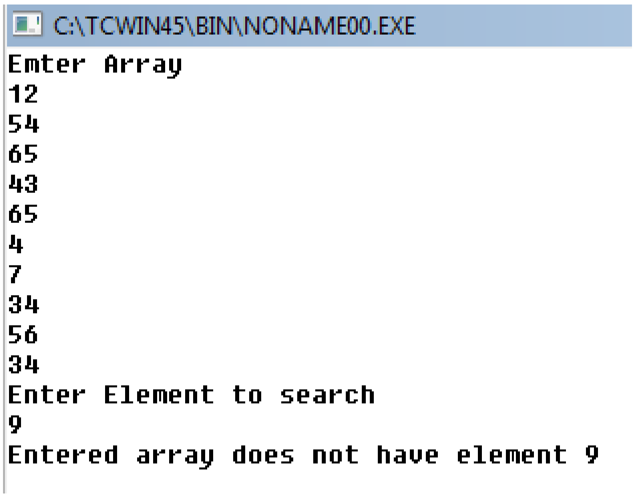
//22

/*WA Function to search whether a float element DATA is present in a sorted array A[N] or not (use <u>Linear Search Technique</u>). Also write a minimum program to invoke the function.*/

```
#include<iostream.h>
#include<conio.h>
int Lsearch(float A[],int N,float DATA)
{
        int i,flag=0;
        for(i=0;i<N;i++)
    {
        if(A[i]==DATA)
      flag=1;
    }
        return flag;
}
void main()
{
        float A[10];
        cout<<"Emter Array"<<endl;
        for(int i=0;i<10;i++)
        cin>>A[i];
        cout<<"Enter Element to search"<<endl;
        float DAT;
        cin>>DAT;

        if(Lsearch(A,10,DAT)==1)
        cout<<"Enter array has element "<<DAT;
        else
        cout<<"Entered array does not have element "<<DAT;
        getch();
}
```



```
C:\TCWIN45\BIN\NONAME00.EXE
Emter Array
12
34
56
34
65
23
56
34
23
12
Enter Element to search
56
Enter array has element 56
```

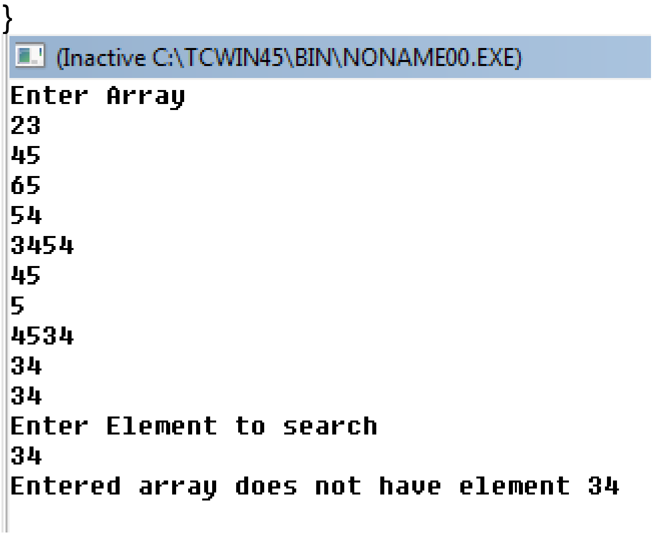


```
C:\TCWIN45\BIN\NONAME00.EXE
Emter Array
12
54
65
43
65
4
7
34
56
34
Enter Element to search
9
Entered array does not have element 9
```

//23

/*WA Function to search whether a float element DATA is present in a sorted array A[N] or not (use Binary Search Technique). Also write a minimum program to invoke the function.*/

```
#include<iostream.h>
#include<conio.h>
int Bsearch(float A[],int N,float DATA)
{
        int i,flag=0;
        int L=0,U=N-1,M;
        while(flag==0&&U>=L)
        {
                M=(U+L)/2;
                if(A[M]==DATA)
                flag=1;
                else if(A[M]>DATA)
                U=M-1;
                else
                L=M+1;
        }
        return flag;
}
void main()
{
        float A[10];
        cout<<"Enter Array"<<endl;
        for(int i=0;i<10;i++)
        cin>>A[i];
        cout<<"Enter Element to search"<<endl;
        float DAT;
        cin>>DAT;

        if(Bsearch(A,10,DAT)==1)
        cout<<"Enter array has element "<<DAT;
        else
        cout<<"Entered array does not have element "<<DAT;
        getch();
}
```



```
Enter Array
23
45
65
54
3454
45
5
4534
34
34
Enter Element to search
34
Entered array does not have element 34
```

//24

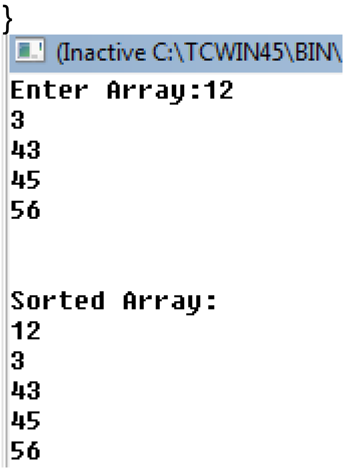
/*WAF to accept an array A[N] of float numbers and then return the same array but in ascending order. (Use Linear Sort Technique). Also write a minimum C++ program to illustrate the defined function.*/

```
#include<iostream.h>
#include<conio.h>
void seq_sort(float A[],int N)
{
        int i,j;
        float T;
        for(i=0;i<N-1;i++)
        {
                for(j=i+1;j<N;j++)
                {
                        if(A[i]>A[j])
                        {
                                T=A[j];
                                A[j]=A[i];
                                A[i]=T;
                        }
                }
        }
}
void main()
{
        float X[5];
        cout<<"Enter Array:";
        for(int i=0;i<5;i++)
        cin>>X[i];
        seq_sort(X,5);
        cout<<"\n\nSorted Array:\n";
        for(int i=0;i<5;i++)
        cout<<X[i]<<endl;
        getch();
}
```

//25

/*WAF to accept an array A[N] of float numbers and then return the same array but in ascending order. (Use Selection Sort Technique). Also write a minimum C++ program to illustrate the defined function.*/

```
#include<iostream.h>
#include<conio.h>
void sel_sort(float A[],int N)
{
	int i,j,M;
	float T;
	for(i=0;i<N-1;i++)
	{
		M=i;
		for(j=i+1;j<N;j++)
		{
			if(A[M]>A[j])
			M=j;
			if(M!=i)
			{
				T=A[M];
				A[M]=A[i];
				A[i]=T;
			}
		}
	}
}
void main()
{
	float X[5];
	cout<<"Enter Array:";
	for(int i=0;i<5;i++)
	cin>>X[i];
	sel_sort(X,5);
	cout<<"\n\nSorted Array:\n";
	for(int i=0;i<5;i++)
	cout<<X[i]<<endl;
	getch();
}
```



```
Enter Array:12
3
43
45
56


Sorted Array:
12
3
43
45
56
```

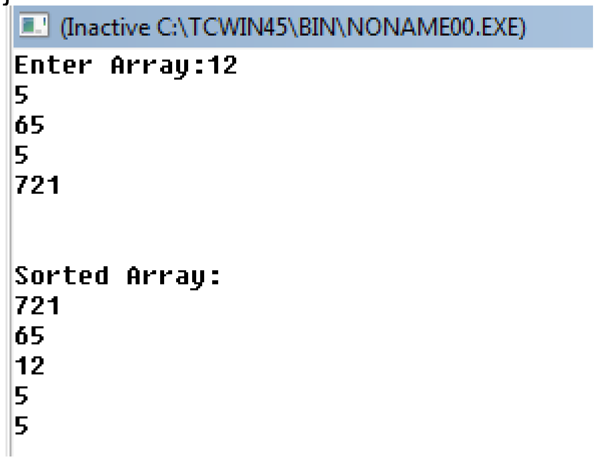
//26
/*WAF to accept an array A[N] of float numbers and then return the same array but in descending order. (Use Bubble Sort Technique). Also write a minimum C++ program to illustrate the defined function.*/

```
#include<iostream.h>
#include<conio.h>
void bub_sort(float A[],int N)
{
        int i,j;
        float T;
        for(i=0;i<N-1;i++)
        {
                for(j=0;j<N-1-i;j++)
                {
                        if(A[j]<A[j+1])
                        {
                                T=A[j];
                                A[j]=A[j+1];
                                A[j+1]=T;
                        }
                }
        }
}
void main()
{
        float X[5];
        cout<<"Enter Array:";
        for(int i=0;i<5;i++)
        cin>>X[i];
        bub_sort(X,5);
        cout<<"\n\nSorted Array:\n";
        for(int i=0;i<5;i++)
        cout<<X[i]<<endl;
        getch();
}
```

//27

/*WAF to accept an array A[N] of float numbers and then return the same array but in descending order. (Use <u>Insertion Sort Technique</u>). Also write a minimum C++ program to illustrate the defined function.*/

```
#include<iostream.h>
#include<conio.h>
void ins_sort(float A[],int N)
{
        int i,j,k;
        float T;
        for(i=0;i<N;i++)
        {
                T=A[0];
                for(j=N-1;j>=N-i&&A[j]<T;j--);
                for(k=0;k<j;k++)
                A[k]=A[k+1];
                A[j]=T;
        }
}
void main()
{
        float X[5];
        cout<<"Enter Array:";
        for(int i=0;i<5;i++)
        cin>>X[i];
        ins_sort(X,5);
        cout<<"\n\nSorted Array:\n";
        for(int i=0;i<5;i++)
        cout<<X[i]<<endl;
        getch();
}
```



(Inactive C:\TCWIN45\BIN\NONAME00.EXE)

```
Enter Array:12
5
65
5
721


Sorted Array:
721
65
12
5
5
```

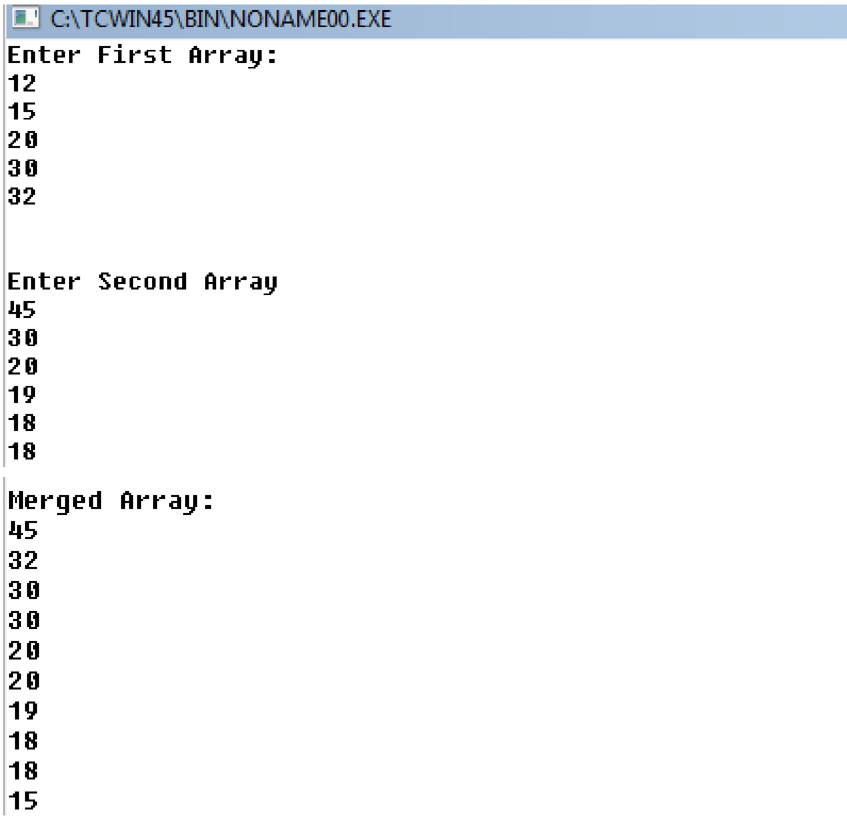
//28
/*WAF to accept two arrays A[M] and B[N] and then return the merged array C[]. (All the arrays are in ascending order).*/

```
#include<iostream.h>
#include<conio.h>
void merge(float A[],int N,float B[],int M,float C[])
{
        int i=0,j=0,k=0;
        while(i<N&&j<M)
  {
        if(A[i]<B[j])
    C[k++]=A[i++];

    else
    C[k++]=B[j++];
  }
  while(i<N)
  {
        C[k++]=A[i++];
  }
  while(j<M)
  {
        C[k++]=B[j++];
  }
}
void main()
{
        float X[5];
  float Z[11];
  for(int i=0;i<5;i++)
  cin>>X[i];
  cout<<endl;
  cout<<endl;
  float Y[6];
  for(int i=0;i<6;i++)
  cin>>Y[i];
  cout<<endl;
  cout<<endl;
  merge(X,5,Y,6,Z);
  for(int i=0;i<11;i++)
  cout<<Z[i]<<endl;
  getch();
}
```

//29

/*WAF to accept two arrays A[M] and B[N] and then return the merged array C[]. (where A is in ascending order and B is in descending order. C should be in descending order)*/

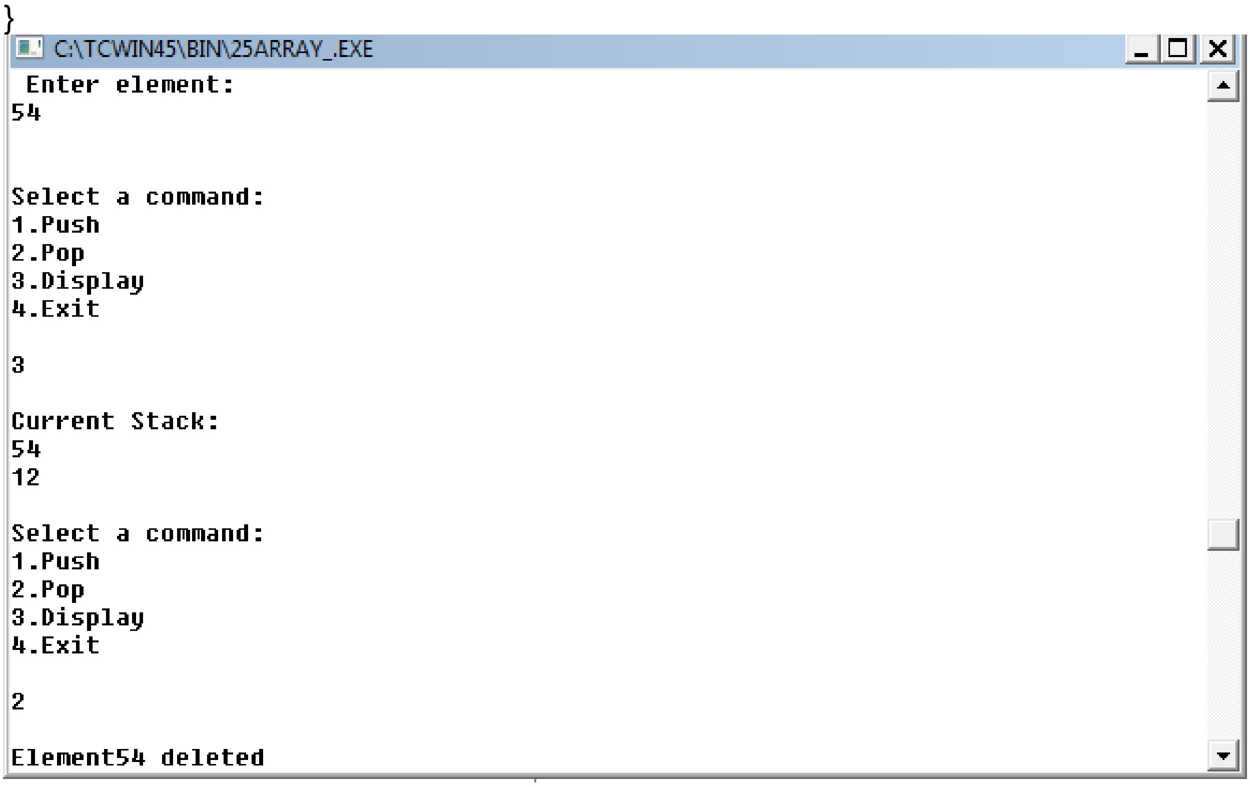
```
#include<iostream.h>
#include<conio.h>
void merge(float A[],int N,float B[],int M,float C[])
{
        int i=N-1,j=0,k=0;
        while(i>0&&j<M)
  {
        if(A[i]>B[j])
     C[k++]=A[i--];

     else
     C[k++]=B[j++];
  }
  while(i>0)
  {
        C[k++]=A[i--];
  }
  while(j<M)
  {
        C[k++]=B[j++];
  }
}
void main()
{
        float X[5];
  float Z[11];
  cout<<”Enter First Array:\n”;
  for(int i=0;i<5;i++)
  cin>>X[i];
  cout<<endl<<endl;
  cout<<”Enter Second Array\n”;
  float Y[6];
  for(int i=0;i<6;i++)
  cin>>Y[i];
  cout<<endl<<endl;
  cout<<”Merged Array:\n”;
  merge(X,5,Y,6,Z);
  for(int i=0;i<11;i++)
  cout<<Z[i]<<endl;
  getch();
}
```



```
Enter First Array:
12
15
20
30
32


Enter Second Array
45
30
20
19
18
18

Merged Array:
45
32
30
30
20
20
19
18
18
15
```

//30
/*WA menu based OOP for ARRAY implementation of STACK.*/

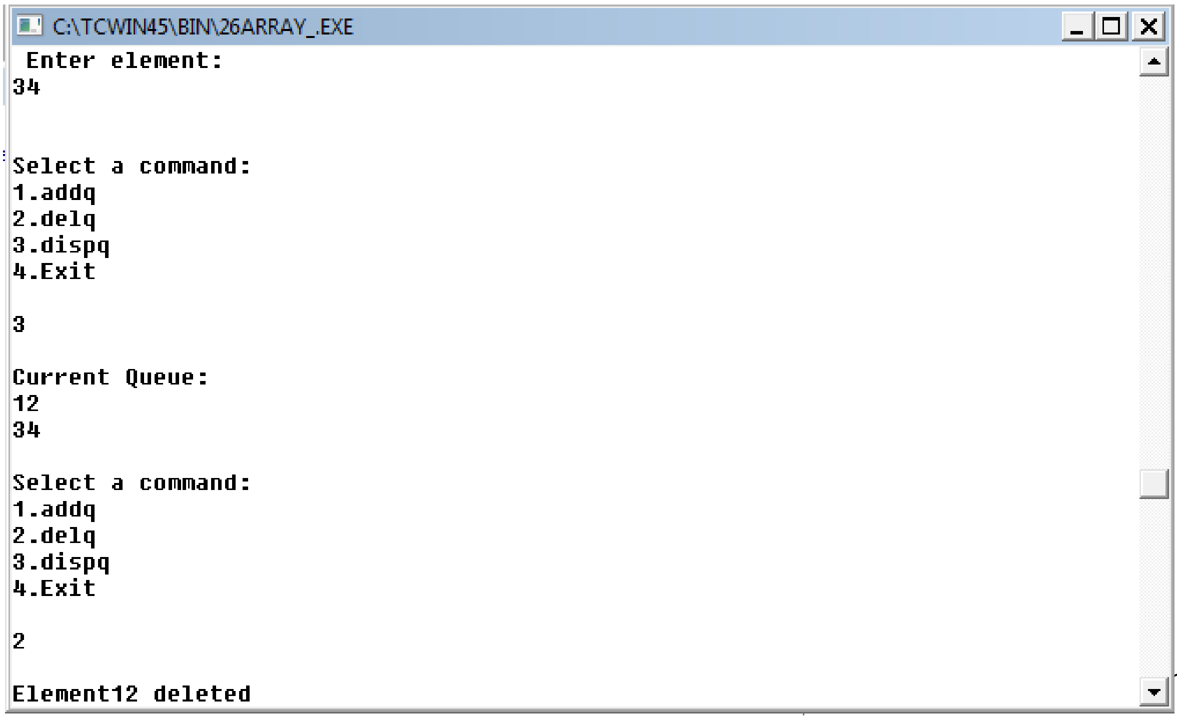
```
#include<iostream.h>
#include<conio.h>
const N=3;
class Stack
{
                float A[N];
                int top;
        public:
                Stack()
                {
                        top=-1;
                }

                void Push();
                void Pop();
                void Display();
};

void Stack::Push()
{
        if(top==N-1)
                cout<<"\n\nOverflow! \n addition not possible.\n\n";
        else
        {
                top++;
                cout<<"\n\n Enter element:\n";
                cin>>A[top];
        }
}

void Stack::Pop()
{
        if(top==-1)
                cout<<"\n\nUnderflow! Stack is empty; Deletion not possible.\n\n";

        else
        {
                cout<<"\n\nElement"<<A[top]<<" deleted\n\n";
                top--;
        }
}

void Stack::Display()
{
        if(top==-1)
                cout<<"\n\nStack is empty\n\n";
        else
        {
                cout<<"\n\nCurrent Stack:";
                for(int i=top;i>=0;i--)    //Displays Array in reverse order so that
                        cout<<endl<<A[i];       //Top ends up on top
        }
}


void main()
{
        char choice;
        Stack S;
        do
        {
                cout<<"\n\nSelect a command:\n1.Push\n2.Pop\n3.Display\n4.Exit\n\n";
                choice=getche();
                switch(choice)
                {
                        case'1':        S.Push();
                                        break;
                        case'2':        S.Pop();
                                        break;
                        case'3':        S.Display();
                                        break;
                        case'4':
                                        break;
                        default:        cout<<"Invalid Choice!\n\n";
                }
        }while(choice!='4');
}
```



```
 Enter element:
54


Select a command:
1.Push
2.Pop
3.Display
4.Exit

3

Current Stack:
54
12

Select a command:
1.Push
2.Pop
3.Display
4.Exit

2

Element54 deleted
```

```
//31
/*WA menu based OOP for ARRAY implementation of QUEUE.
*/

#include<iostream.h>
#include<conio.h>
const N=3;

class Queue
{
                float A[N];
                int rear;
        public:
                Queue()
                {
                        rear=-1;
                }

                void addq();
                void delq();
                void dispq();
};

void Queue::addq()
{
        if(rear==N-1)
                cout<<"\n\nOverflow! \n addition not possible.\n\n";
        else
        {
                rear++;
                cout<<"\n\n Enter element:\n";
                cin>>A[rear];
        }
}

void Queue::delq()
{
        if(rear==-1)
                cout<<"\n\nUnderflow! Queue is empty; Deletion not possible.\n\n";

        else
        {
                cout<<"\n\nElement"<<A[0]<<" deleted\n\n";
                rear--;
                for(int i=0;i<N;i++)
                        A[i]=A[i+1];
        }
}
```

```
void Queue::dispq()
{
        if(rear==-1)
                cout<<"\n\nQueue is empty\n\n";
        else
        {
                cout<<"\n\nCurrent Queue:";
                for(int i=0;i<=rear;i++)
                        cout<<endl<<A[i];
        }
}


void main()
{
        char choice;
        Queue S;
        do
        {
                cout<<"\n\nSelect a command:\n1.addq\n2.delq\n3.dispq\n4.Exit\n\n";
                choice=getche();
                switch(choice)
                {
                        case'1':        S.addq();
                                        break;
                        case'2':        S.delq();
                                        break;
                        case'3':        S.dispq();
                                        break;
                        case'4':
                                        break;
                        default:        cout<<"Invalid Choice!\n\n";
                }
        }while(choice!='4');
}
```



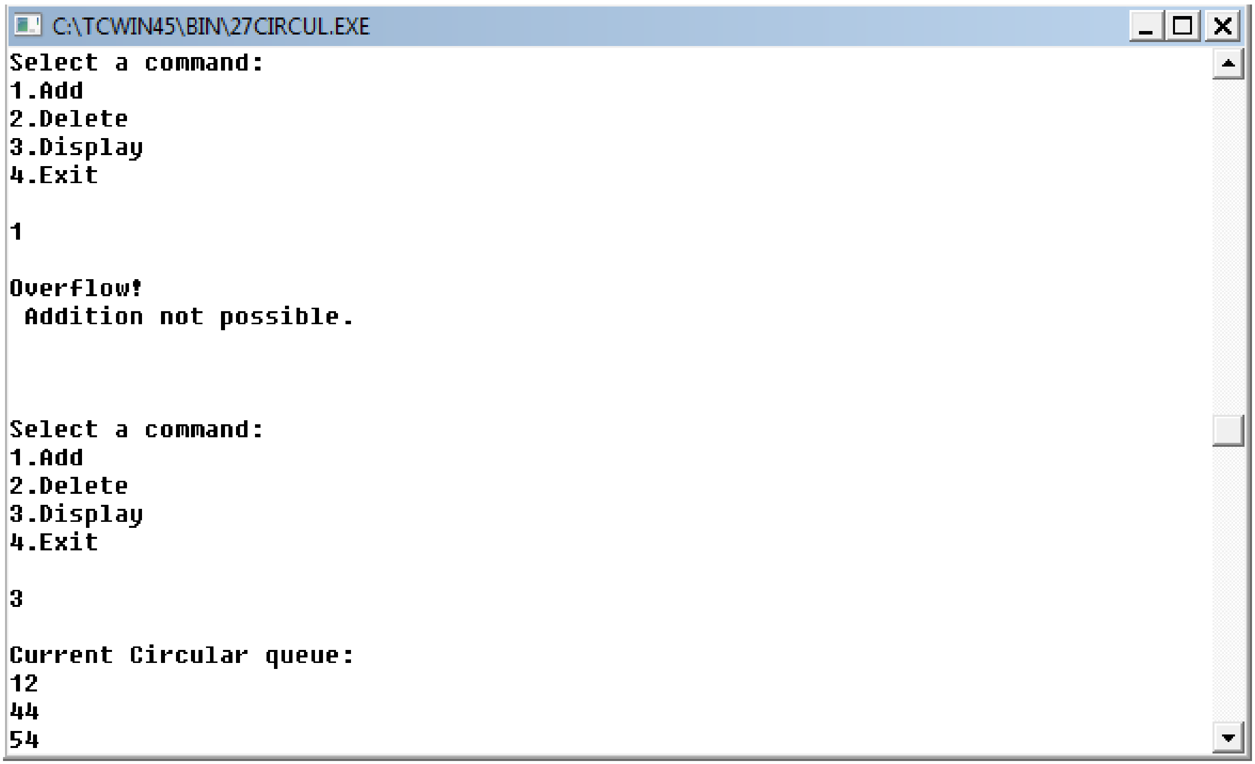
```
 Enter element:
34


Select a command:
1.addq
2.delq
3.dispq
4.Exit

3

Current Queue:
12
34

Select a command:
1.addq
2.delq
3.dispq
4.Exit

2

Element12 deleted
```

//32
/*WA menu based OOP for ARRAY implementation of CIRCULAR QUEUE
***/**

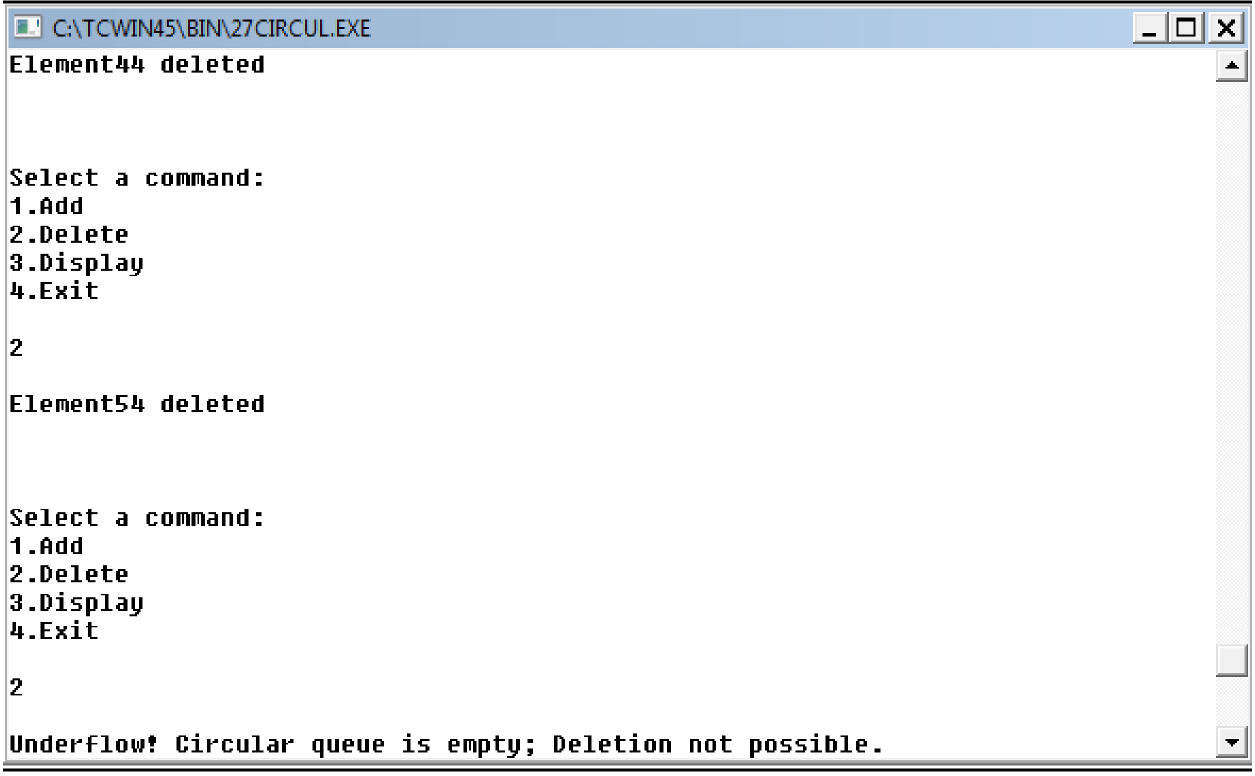
```
#include<iostream.h>
#include<conio.h>
const N=3;

class Cqueue
{
		float A[N];
		int front;
		int rear;
	public:
		Cqueue()
		{
			front=-1;
			rear=-1;
		}

		void addCq();
		void delCq();
		void dispCq();
};

void Cqueue::addCq()
{
	if((front==0&rear==N-1)||(front==rear+1))
		cout<<"\n\nOverflow! \n Addition not possible.\n\n";
	else
	{
		rear++;
		if(rear==N)
			rear=0;
		cout<<"\n Enter element:\n";
		cin>>A[rear];
		if(front==-1)
			front=0;
	}
}

void Cqueue::delCq()
{
	if(rear==-1)
		cout<<"\n\nUnderflow! Circular queue is empty; Deletion not
possible.\n\n";

	else
	{
		cout<<"\n\nElement"<<A[front]<<" deleted\n\n";
```

```
                if(front==rear)
                {
                        rear=-1;
                        front=-1;
                }
                else
                {
                        front++;
                        if(front==N)
                                front=0;
                }
        }
}

void Cqueue::dispCq()
{
        if(rear==-1)
                cout<<"\n\nCqueue is empty\n\n";
        else
        {
                cout<<"\n\nCurrent Circular queue:";

                if(front<=rear)
                {
                        for(int i=front;i<=rear;i++)
                                cout<<endl<<A[i];
                }
                else
                {
                        for(int k=front;k<N;k++)
                                cout<<endl<<A[k];
                        for(int j=0;j<=rear;j++)
                                cout<<endl<<A[j];
                }
        }
}


void main()
{
        char choice;
        Cqueue S;
        do
        {
                cout<<"\n\nSelect a command:\n1.Add
\n2.Delete\n3.Display\n4.Exit\n\n";
                choice=getche();
                switch(choice)
                {
                        case'1':        S.addCq();
                                        break;
                        case'2':        S.delCq();
```

```
                                        break;
                        case'3':        S.dispCq();
                                        break;
                        case'4':
                                        break;
                        default:        cout<<"Invalid Choice!\n\n";
                }
        }while(choice!='4');
}
```



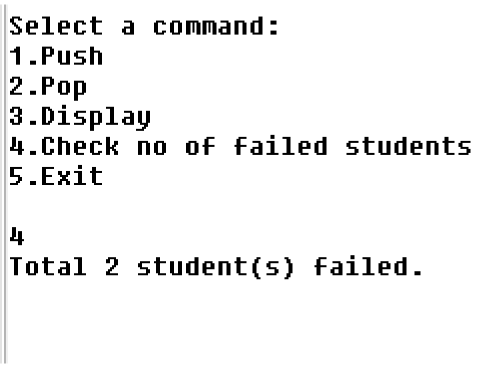
```
Select a command:
1.Add
2.Delete
3.Display
4.Exit

1

Overflow!
 Addition not possible.



Select a command:
1.Add
2.Delete
3.Display
4.Exit

3

Current Circular queue:
12
44
54
```



```
Element44 deleted



Select a command:
1.Add
2.Delete
3.Display
4.Exit

2

Element54 deleted



Select a command:
1.Add
2.Delete
3.Display
4.Exit

2

Underflow! Circular queue is empty; Deletion not possible.
```

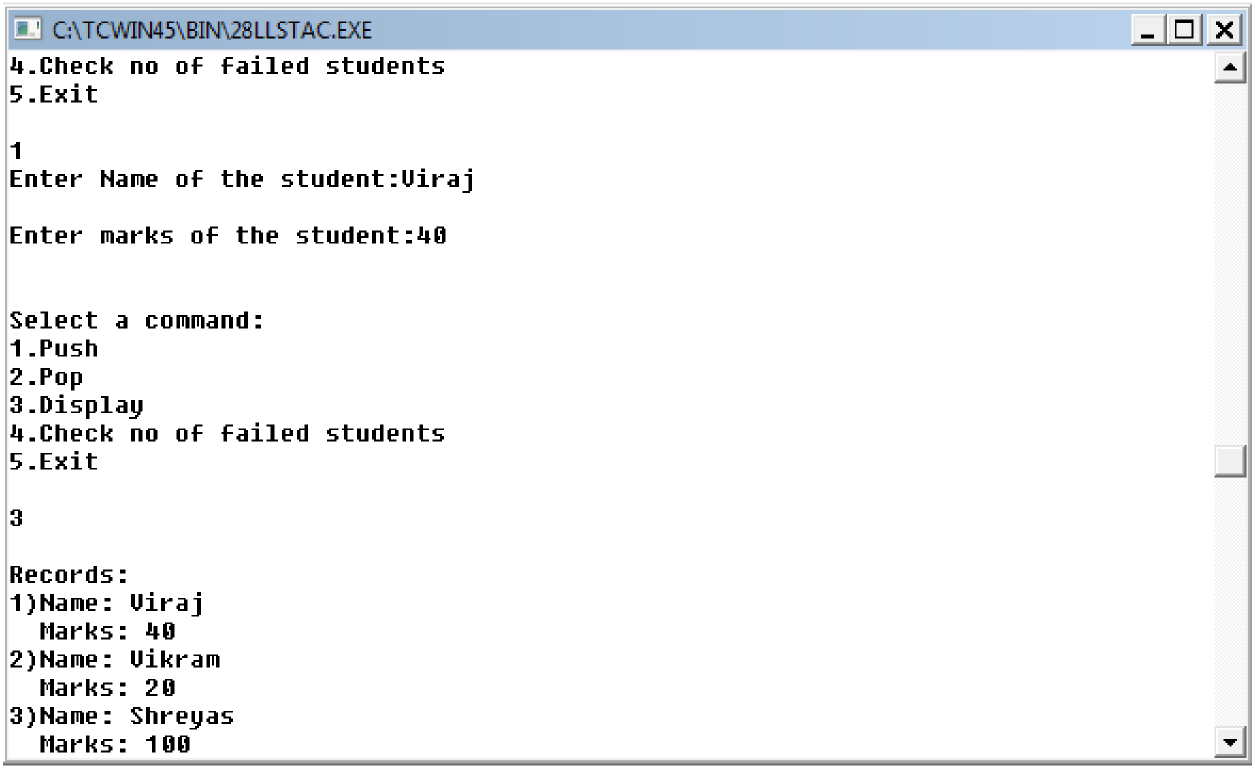
//33
/*WAP (menu based) for Linked list implementation of STACK, where each node consist of name & marks of students.
Also define a (member) function to count total number of students failed (marks<40).
***/**

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

struct node
{
        char name[20];
        int marks;
        node *next;
};

class Stack
{
                node *top;
        public:
                Stack()
                {
                        top=NULL;
                }
                void Push();
                void Pop();
                void Display();
                ~Stack();
                int FailNo();
};

void Stack::Push()
{
        node *temp;
        temp=new node;
        cout<<"\nEnter Name of the student:";
        gets(temp->name);
        cout<<"\nEnter marks of the student:";
        cin>>temp->marks;
        temp->next=top;
        top=temp;
}

void Stack::Pop()
{
        if(top==NULL)
                cout<<"\n\nUnderflow! Stack is empty\n\n";
        else
        {
                node *temp;
                temp=top;
                top=top->next;
    cout<<"\nRecord of "<<temp->name<<" deleted\n";
                delete(temp);
        }
}

void Stack::Display()
{
        if(top==NULL)
                cout<<"\n\n Stack is empty.\n\n";
        else
        {
                node *temp;
                temp=top;
                int count=1;

                cout<<"\n\nRecords:\n";

                while(temp!=NULL)
                {
                        cout<<count++<<")";
                        cout<<"Name: "<<temp->name<<endl;
                        cout<<"  Marks: "<<temp->marks<<endl;
      temp=temp->next;
                }
        }
}

Stack::~Stack()
{
        node *temp;
        while(top!=NULL)
        {
                temp=top;
                top=top->next;
                delete(temp);
        }
}

int Stack::FailNo()
{
        node *temp;
        temp=top;
        int count=0;
        while(temp!=NULL)
        {
                if(temp->marks<=40)
                        count++;
                temp=temp->next;
```



```
Select a command:
1.Push
2.Pop
3.Display
4.Check no of failed students
5.Exit

4
Total 2 student(s) failed.
```

```
        }
        return count;
}

void main()
{
        char choice;
        Stack S;
        do
        {
                cout<<"\n\nSelect a command:\n1.Push \n2.Pop\n3.Display\n4.Check no
of failed students \n5.Exit\n\n";
                choice=getche();
                switch(choice)
                {
                        case'1':        S.Push();
                                        break;
                        case'2':        S.Pop();
                                        break;
                        case'3':        S.Display();
                                        break;
                        case'4':        cout<<"\nTotal "<<S.FailNo()<<" student(s) failed.\n\n";
                                        break;
                        case'5':        break;
                        default:        cout<<"\nInvalid Choice!\n\n";
                }
        }while(choice!='5');
}
```

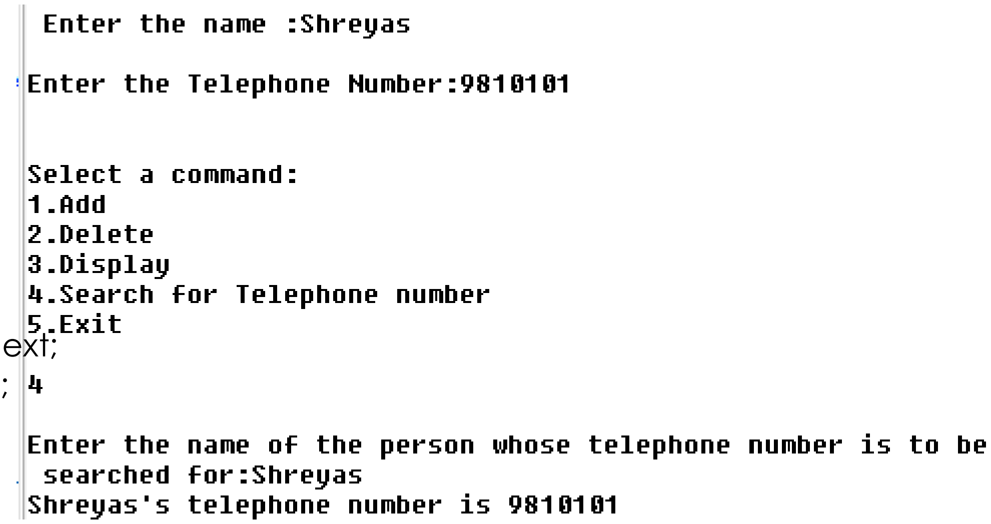


```
4.Check no of failed students
5.Exit

1
Enter Name of the student:Viraj

Enter marks of the student:40


Select a command:
1.Push
2.Pop
3.Display
4.Check no of failed students
5.Exit

3

Records:
1)Name: Viraj
  Marks: 40
2)Name: Vikram
  Marks: 20
3)Name: Shreyas
  Marks: 100
```

```
//34
/*WA menu based program for Linked list implementation of QUEUE, where each node
consist of name and telephone number of a person.
Also define a (member) function to search telephone number of a person.
*/

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

struct node
{
    char name[20];
    long telno;
    node *next;
};

class Queue
{
        node *front;
        node *rear;
    public:
        Queue()
        {
            front=rear=NULL;
        }

        void addq();
        void delq();
        void display();
        void telsearch();
        ~Queue();
};

void Queue::addq()
{
    node *temp;
    temp=new node;
    cout<<"\n\n Enter the name :";
    gets(temp->name);
    cout<<"\nEnter the Telephone Number:";
    cin>>temp->telno;
    temp->next=NULL;
    if(front!=NULL)
    {
        rear->next=temp;
        rear=temp;
    }
    else
```

```
		front=rear=temp;
	}

	void Queue::delq()
	{
		if(front==NULL)
			cout<<"\n\nUnderflow! Queue is empty.\n\n";
		else
		{
			node *temp;
			temp=front;
			front=front->next;
			cout<<"\n\nRecord of "<<temp->name<<" deleted.\n\n";
			delete(temp);
			if(front==NULL)
				rear=NULL;
		}
	}

	void Queue::display()
	{
		if(front==NULL)
			cout<<"\n\nQueue is empty.\n\n";
		else
		{
			node *temp;
			temp=front;
			int count=1;
			cout<<"\n\nRecords:\n";
			while(temp!=NULL)
			{
				cout<<endl<<endl<<count++;
				cout<<"		Name:"<<temp->name;
				cout<<"\nTelephone Number:"<<temp->telno;
				temp=temp->next;
			}
		}
	}

	void Queue::telsearch()
	{
		cout<<"\n\nEnter the name of the person whose telephone number is to be”;
		cout<<”\nsearched for:";
		char Name[20];
		gets(Name);
		node *temp;
		temp=front;
		int count=0;
		while(temp!=NULL)
			{
				if(strcmp(Name,temp->name)==0)
				{
```

```
                count++;
                cout<<Name<<"'s telephone number is "<<temp->telno<<endl;
            }
            temp=temp->next;
        }
    if(count==0)
        cout<<"\n\nName not found!\n\n";
}

Queue::~Queue()
{
    node *temp;

    while(front!=NULL)
    {
        temp=front;
        front=front->next;
        delete(temp);
    }
}
```



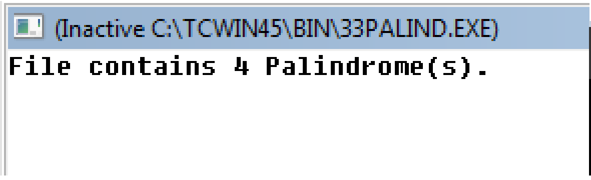
```
 Enter the name :Shreyas

Enter the Telephone Number:9810101


Select a command:
1.Add
2.Delete
3.Display
4.Search for Telephone number
5.Exit

4

Enter the name of the person whose telephone number is to be
 searched for:Shreyas
Shreyas's telephone number is 9810101
```

```
void main()
{
    char choice;
    Queue S;
    do
    {
        cout<<"\n\nSelect a command:\n1.Add \n2.Delete\n3.Display\n4.Search
for Telephone number \n5.Exit\n\n";
        choice=getche();
        switch(choice)
        {
            case'1':    S.addq();
                        break;
            case'2':    S.delq();
                        break;
            case'3':    S.display();
                        break;
            case'4':    S.telsearch();
                        break;
            case'5':    break;
            default:    cout<<"Invalid Choice!\n\n";
        }
    }while(choice!='5');
}
```

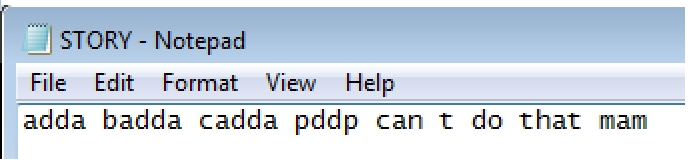
```
/*35
WAP to read sentences through the (user)
key board and save those into a text file DATA.TXT
*/

#include<fstream.h>

void main()
{
        char text[80];
        cout<<"Enter text\n";
        fstream f1;
        f1.open("DATA.TXT",ios::out);
        cin.getline(text,80,'.');    /*max length of line is 80 characters and
                                                                        '.' is terminating
character*/
        f1<<text;
}

//36
/*WAP to read a text file STORY.TXT  and count the number of sentences, words,
uppercase characters, lowercase characters, digits and special characters
in that file.
*/

#include<fstream.h>
#include<conio.h>
#include<ctype.h>

void main()
{
        int linec=0;
        int word=0;
        int uchar=0;
        int lchar=0;
        int digit=0;
        int splchar=0;
        ifstream fin("story.txt");

        char line[255];
        char c;

        while(!fin.eof())
        {
                fin.getline(line,255,'.');              //Max length of line=255char . is terminating
                linec++;
        }

        fin.seekg(0,ios::beg);

        while(!fin.eof())
        {
                fin>>line;
                word++;
        }

        fin.seekg(0);

        while(!fin.eof())
        {
                fin>>c;
                if(isupper(c))
                        uchar++;
        }
        fin.seekg(0);
        while(!fin.eof())
        {
                fin>>c;
                if(islower(c))
                        lchar++;
        }
        fin.seekg(0);
        while(!fin.eof())
        {
                fin>>c;
                if(isdigit(c))
                        digit++;
        }
        fin.seekg(0);
        while(!fin.eof())
        {
                fin>>c;
                if(isalnum(c)==0)
                        splchar++;
        }

        cout<<" Number of Lines in file= "<<linec<<".\n";
        cout<<" Number of Words in file= "<<word<<".\n";
        cout<<" Number of Upper case characters in file= "<<uchar<<".\n";
        cout<<" Number of lower case characters in file= "<<lchar<<".\n";
        cout<<" Number of special characters in file= "<<splchar<<".\n";
}
```

```
//36
/*WAP to read a text file MESSAGE.TXT  and modify that file by converting all characters
(or words) of that file into UPPERCASE characters ( or  words).
*/
#include<fstream.h>
#include<stdio.h>
#include<ctype.h>

void main()
{
	char w;

	fstream f1("story.txt",ios::in);
	fstream f2("temp.txt",ios::out);

	while(!f1.eof())
	{
		f1>>w;
		if(islower(w))
			w=toupper(w);

		f2<<w;
	}
	f1.close();
	f2.close();
	remove("story.txt");
	rename("temp.txt","story.txt");


}
```

```
//37
/*WA Function to accept a string as argument and returns 1 if the string is palindrome otherwise 0.
WAP to read a text file STORY.TXT and calculate the total number of palindrome the file contains.
*/

#include<fstream.h>
#include<string.h>

int palindrome(char word[])
{
        int L=strlen(word);
        int flag=1;
        for(int i=0;i<=L/2;i++)
        {
                if(word[i]!=word[L-i-1])
                        flag=0;
        }
        return flag;
}


void main()
{
        ifstream fin("story.txt");
        char word[40];
        int count=0;
        while(!fin.eof())
        {
                fin>>word;
                if(palindrome(word))
                        count++;
        }
  fin.close();
        cout<<"File contains "<<count<<" Palindrome(s).";
}
```



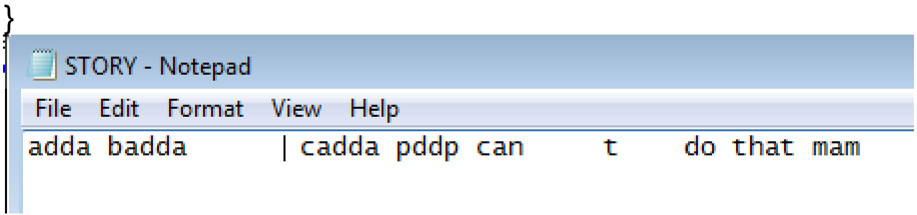
```
File contains 4 Palindrome(s).
```



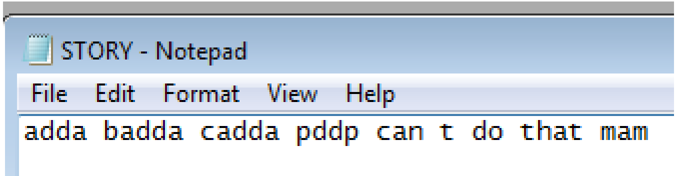
```
adda badda cadda pddp can t do that mam
```

//38
/*WAP to read a text file STORY.TXT and modify that file by deleting all repeating spaces between words.
The original file contains multiple spaces between few words.
Your program will remove all multiple spaces such that the number of space between two words must be ONE.
***/**

```
#include<fstream.h>
#include"stdio.h"
#include<string.h>

void main()
{
        char word[20];

        fstream f1("story.txt",ios::in);
        fstream f2("temp.txt",ios::out);

        while(!f1.eof())
        {
                f1>>word;
                f2<<word<<" ";
        }
        f1.close();
        f2.close();
        remove("story.txt");
        rename("temp.txt","story.txt");

}
```

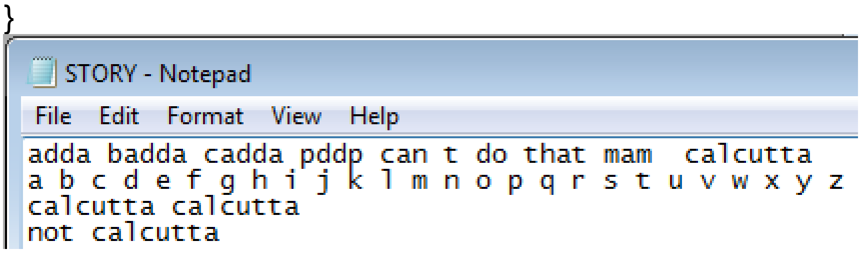


```
adda badda      | cadda pddp can     t     do that mam
```

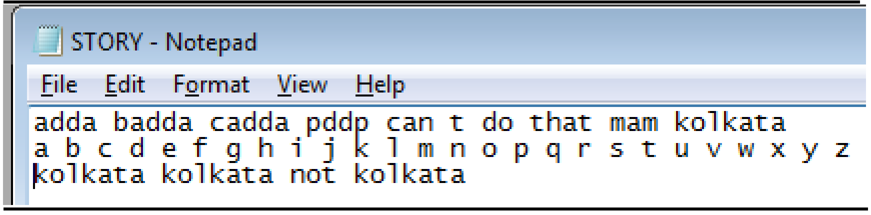


```
adda badda cadda pddp can t do that mam
```

```
//39
/*WAP to read a text file STORY.TXT modify the file by replacing all 'calcutta' with
'kolkata'.
*/
#include<fstream.h>
#include"stdio.h"
#include<string.h>

void main()
{
	char word[20];

	fstream f1("story.txt",ios::in);
	fstream f2("temp.txt",ios::out);

	while(!f1.eof())
	{
		f1>>word;
		if(strcmpi(word,"calcutta")==0)

		f2<<"kolkata ";

		else
		f2<<word<<" ";
	}
	f1.close();
	f2.close();
	remove("story.txt");
	rename("temp.txt","story.txt");

}
```



```
adda badda cadda pddp can t do that mam  calcutta
a b c d e f g h i j k l m n o p q r s t u v w x y z
calcutta calcutta
not calcutta
```



```
adda badda cadda pddp can t do that mam kolkata
a b c d e f g h i j k l m n o p q r s t u v w x y z
kolkata kolkata not kolkata
```

//40
/*WA menu based OOP to SEARCH, DISPLAY, ADD, DELETE and MODIFY some records stored in a binary file TEL.DAT.
A record consists of name (20 char) and Telephone number (10 char).
***/**

```
#include<fstream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>


class Person
{
                char Name[50];
                long Telno;
                char Email[30];
        public:
                void Getdata();
                void Showdata();
                int  CheckN(char *);
                int  CheckT(long);
                int  CheckE(char *);
                void Edit();
};



//      **************************** INPUT AN OBJECT *****************************
void Person::Getdata()
{
        cout<<"\n\nEnter Name : ";
        gets(Name);
        cout<<"Enter Telephone number : ";
        cin>>Telno;
        cout<<"Enter Email ID : ";
        gets(Email);
}



//      **************************** DISPLAY AN OBJECT ****************************
void Person::Showdata()
{
        cout<<"\n\nName : "<<Name<<endl;
        cout<<"Telephone number : "<<Telno<<endl;
        cout<<"Email ID : "<<Email<<endl<<endl;
}
//      ******************* CHECK NAME FOR AN OBJECT ************************
```

```
int Person::CheckN(char name[])
{

        if (strcmp(name,Name)==0)
                return 1;
        else
                return 0;
}





//      ************ CHECK TELEPHONE NUMBER FOR AN OBJECT ***************
int Person::CheckT(long tel)
{
        if (tel==Telno)
                return 1;
        else
                return 0;
}





//      ******************** CHECK EMAIL ID FOR AN OBJECT **********************
int Person::CheckE(char mail[])
{
        if (strcmp(mail,Email)==0)
                return 1;
        else
                return 0;
}



//      ************************* EDIT/ MODIFY AN OBJECT ************************
void Person::Edit()
{
        char choice;
        do{
                cout<<"Current detail of the Record :\n\n";
                Showdata();
                cout<<"\n\nPress A if you want to change Name \n";
                cout<<"Press B if you want to change Telephone Number ID \n";
                cout<<"Press C if you want to change Email ID \n";
                cout<<"Press D if you want no more change \n";
                cout<<"Enter your option :";
                choice=getche();
```

```
		switch(choice)
		{
			case 'A':
			case 'a':	cout<<"\nEnter new Name : ";
						gets(Name);
						break;
			case 'B':
			case 'b':	cout<<"\nEnter new Telephone Number : ";
						cin>>Telno;
						break;
			case 'C':
			case 'c':	cout<<"\nEnter new Email ID : ";
						gets(Email);
						break;
			case 'D':
			case 'd':	break;

			default :	cout<<"\nInvalid Choice !! Try Again !!";
		}
	}while(choice!='D'&&choice!='d');
}




//	******************** APPEND (ADD) A NEW RECORD ***********************
void Append()
{

	Person P;
	cout<<"\nEnter the Detail of the Record : \n";
	P.Getdata();

	fstream f1;
	f1.open("Tel.dat",ios::binary|ios::app);

	f1.write((char*)&P,sizeof(P));

	f1.close();

}




//	*********************** DISPLAY ALL RECORDS **************************
void DisplayAll()
{

	Person P;
```

```
        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while(f1.read((char*)&P,sizeof(P)))
                P.Showdata();

        f1.close();

}


//      *********************** SEARCH A RECORD BY NAME *****************************
void SearchN()
{

        char sn[50];
        Person P;
        int Found = 0;

        cout<<"\n\nEnter the Name to be searched : ";
        gets(sn);

        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while(f1.read((char*)&P,sizeof(P)))
        {
                if(P.CheckN(sn)==1)
                {
                        Found++ ;
                        P.Showdata();

                }
        }

        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<<Found<< " RECORD(S) FOUND !! \n\n";
}




//      ***************** SEARCH A RECORD BY TELEPHONE NUMBER *******************
void SearchT()
{
        long stn;
        Person P;
        int Found = 0;

        cout<<"\n\nEnter the Telephone Number to be searched : ";
        cin>>stn;

        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while( f1.read((char*) &P, sizeof(P)) )
        {
                if(P.CheckT(stn)==1)
                {
                        Found++ ;
                        P.Showdata();

                }
        }

        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}






//      *************************** SEARCH A RECORD BY EMAIL ID ***************************
void SearchE()
{

        char sem[50];
        Person P;
        int Found = 0;

        cout<<"\n\nEnter the Email ID to be searched : ";
        gets(sem);

        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while( f1.read((char*) &P, sizeof(P)) )
        {
          if(P.CheckE(sem)==1)
          {
                    Found++ ;
                    P.Showdata();

          }
     }

     f1.close();

     if (Found == 0)
          cout<<"\n\n NO MATCH FOUND !! \n\n";
            else
          cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}






//       **************************** MODIFY A RECORD *****************************
void Modify()
{

     char sn[50];
     Person P;
     int Modified = 0;

     cout<<"\n\nEnter the Name to be modified : ";
     gets(sn);

     fstream f1,f2;
     f1.open("Tel.dat",ios::binary|ios::in);
     f2.open("TEMP.DAT",   ios::binary|ios::out);

     while(f1.read((char*)&P,sizeof(P)) )
     {
          if(P.CheckN(sn)==1)
          {
                    Modified++ ;
                    P.Edit();
          }
          f2.write((char*)&P,sizeof(P));
     }

     f1.close();
     f2.close();
```

```
        if (Modified==0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                {
                        remove("Tel.dat");
                        rename("TEMP.DAT" , "Tel.dat");
                }
}




//      **************************** DELETE A RECORD ****************************
void Delete()
{
        char sn[50];
        Person P;
        int Deleted = 0;

        cout<<"\n\nEnter the Name to be deleted : ";
        gets(sn);

        fstream f1,f2;
        f1.open("Tel.dat",ios::binary|ios::in);
        f2.open("TEMP.DAT",   ios::binary|ios::out);

        while( f1.read((char*) &P, sizeof(P)) )
        {
                if(P.CheckN(sn)==1)
                        Deleted ++ ;
                          else
                        f2.write((char *) &P, sizeof(P));
        }

        f1.close();
        f2.close();



        if (Deleted == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                {
                        remove("Tel.dat");
                        rename("TEMP.DAT" , "Tel.dat");
                }

}
```

```
void main()
{
    char choice;
    do{
        clrscr();
        cout<<"Menu\n";
        cout<<"Press 1 for APPEND A RECORD \n";
        cout<<"Press 2 for DISPLAY ALL RECORDS \n";
        cout<<"Press 3 for SEARCH BY NAME \n";
        cout<<"Press 4 for SEARCH BY TELEPHONE NUMBER \n";
        cout<<"Press 5 for SEARCH BY EMAIL ID \n";
        cout<<"Press 6 for MODIFY A RECORD \n";
        cout<<"Press 7 for DELETE A RECORD \n";
        cout<<"Press 8 to QUIT \n";
        cout<<"Enter your choice : ";
        choice=getche();
        switch(choice)
        {
            case '1':   Append();
                        break;
            case '2':   DisplayAll();
                        break;
            case '3':   SearchN();
                        break;
            case '4':   SearchT();
                        break;
            case '5':   SearchE();
                        break;
            case '6':   Modify();
                        break;
            case '7':   Delete();
                        break;
            case '8':       break;
            default :   cout<<"\nInvalid Choice Entered !!\n\n";
        }
        cout<<"\n\nPress any key to continue!";
        getch();
    }while(choice!='8');
}
```

//41
/*WA menu based OOP to SEARCH, DISPLAY, ADD, DELETE and MODIFY some records stored in a binary file TEL.DAT.
A record consists of name (20 char) and Telephone number (10 char).
(in case of MODIFICATION, open the file in in-out mode and use seekg(), seekp(), tellg(), tellp())

***/**

```
#include<fstream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>


class Person
{
                char Name[50];
                long Telno;
                char Email[30];
        public:
                void Getdata();
                void Showdata();
                int  CheckN(char *);
                int  CheckT(long);
                int  CheckE(char *);
                void Edit();
};



//      **************************** INPUT AN OBJECT *****************************
void Person::Getdata()
{
        cout<<"\n\nEnter Name : ";
        gets(Name);
        cout<<"Enter Telephone number : ";
        cin>>Telno;
        cout<<"Enter Email ID : ";
        gets(Email);
}



//      **************************** DISPLAY AN OBJECT ****************************
void Person::Showdata()
{
        cout<<"\n\nName : "<<Name<<endl;
        cout<<"Telephone number : "<<Telno<<endl;
        cout<<"Email ID : "<<Email<<endl<<endl;
}
```

```
//	**************************** CHECK NAME FOR AN OBJECT *****************************
int Person::CheckN(char name[])
{

	if (strcmp(name,Name)==0)
		return 1;
	else
		return 0;
}





//	****************** CHECK TELEPHONE NUMBER FOR AN OBJECT *****************
int Person::CheckT(long tel)
{
	if (tel==Telno)
		return 1;
	else
		return 0;
}





//	************************ CHECK EMAIL ID FOR AN OBJECT ********************
int Person::CheckE(char mail[])
{
	if (strcmp(mail,Email)==0)
		return 1;
	else
		return 0;
}



//	************************* EDIT/ MODIFY AN OBJECT *****************************
void Person::Edit()
{
	char choice;
	do{
		cout<<"Current detail of the Record :\n\n";
		Showdata();
		cout<<"\n\nPress A if you want to change Name \n";
		cout<<"Press B if you want to change Telephone Number ID \n";
		cout<<"Press C if you want to change Email ID \n";
		cout<<"Press D if you want no more change \n";
		cout<<"Enter your option :";
```

```
        choice=getche();
        switch(choice)
        {
            case 'A':
            case 'a':       cout<<"\nEnter new Name : ";
                                gets(Name);
                                break;
            case 'B':
            case 'b':       cout<<"\nEnter new Telephone Number : ";
                                cin>>Telno;
                                break;
            case 'C':
            case 'c':       cout<<"\nEnter new Email ID : ";
                                gets(Email);
                                break;
            case 'D':
            case 'd':       break;

            default :       cout<<"\nInvalid Choice !! Try Again !!";
        }
    }while(choice!='D'&&choice!='d');
}




//      **************************** APPEND (ADD) A NEW RECORD *****************************
void Append()
{

    Person P;
    cout<<"\nEnter the Detail of the Record : \n";
    P.Getdata();

    fstream f1;
    f1.open("Tel.dat",ios::binary|ios::app);

    f1.write((char*)&P,sizeof(P));

    f1.close();

}




//      **************************** DISPLAY ALL RECORDS *****************************
void DisplayAll()
{

    Person P;
```

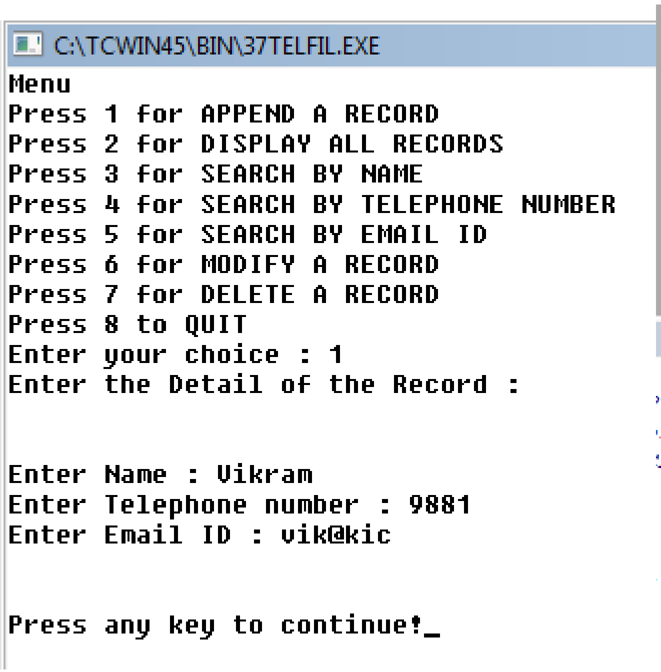
```
        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while(f1.read((char*)&P,sizeof(P)))
                P.Showdata();

        f1.close();

}


//      *********************** SEARCH A RECORD BY NAME **************************
void SearchN()
{

        char sn[50];
        Person P;
        int Found = 0;

        cout<<"\n\nEnter the Name to be searched : ";
        gets(sn);

        fstream f1;
        f1.open("Tel.dat",ios::binary|ios::in);

        while(f1.read((char*)&P,sizeof(P)))
        {
                if(P.CheckN(sn)==1)
                {
                        Found++ ;
                        P.Showdata();

                }
        }

        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<<Found<< " RECORD(S) FOUND !! \n\n";
}
```

```
//    *************** SEARCH A RECORD BY TELEPHONE NUMBER *****************
void SearchT()
{

    long stn;
    Person P;
    int Found = 0;

    cout<<"\n\nEnter the Telephone Number to be searched : ";
    cin>>stn;

    fstream f1;
    f1.open("Tel.dat",ios::binary|ios::in);

    while( f1.read((char*) &P, sizeof(P)) )
    {
        if(P.CheckT(stn)==1)
        {
            Found++ ;
            P.Showdata();

        }
    }

    f1.close();

    if (Found == 0)
        cout<<"\n\n NO MATCH FOUND !! \n\n";
          else
        cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}




//    *************************** SEARCH A RECORD BY EMAIL ID *****************************
void SearchE()
{

    char sem[50];
    Person P;
    int Found = 0;

    cout<<"\n\nEnter the Email ID to be searched : ";
    gets(sem);

    fstream f1;
    f1.open("Tel.dat",ios::binary|ios::in);
```

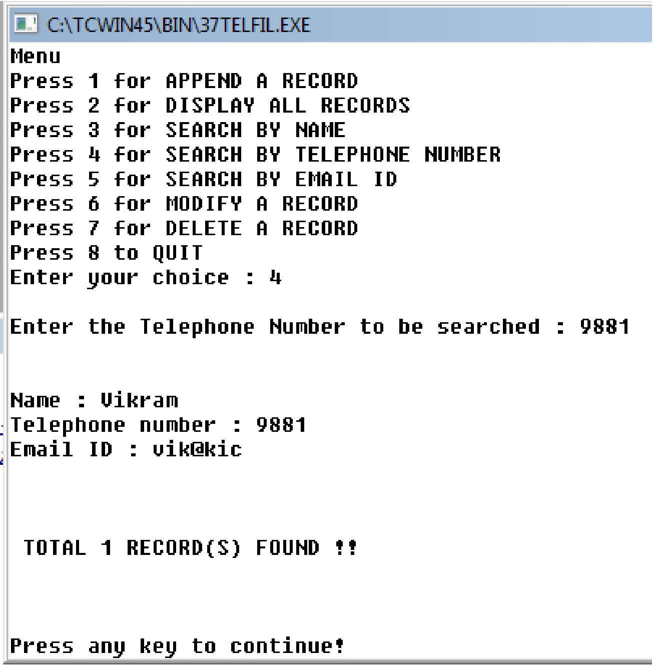
```
	while( f1.read((char*) &P, sizeof(P)) )
	{
		if(P.CheckE(sem)==1)
		{
			Found++ ;
			P.Showdata();

		}
	}

	f1.close();

	if (Found == 0)
		cout<<"\n\n NO MATCH FOUND !! \n\n";
		  else
		cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}






//	*************************** MODIFY A RECORD ****************************
void Modify()
{

	char sn[50];
	Person P;
	int Modified = 0;

	cout<<"\n\nEnter the Name to be modified : ";
	gets(sn);

	fstream f1("Tel.dat",ios::binary|ios::in|ios::out);


	while(f1.read((char*)&P,sizeof(P)) )
	{
		if(P.CheckN(sn)==1)
		{
			Modified++ ;
			P.Edit();
		}
		f1.write((char*)&P,sizeof(P));
	}

	f1.close();
```

```
        if (Modified==0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
}




//      **************************** DELETE A RECORD *****************************
void Delete()
{
        char sn[50];
        Person P;
        int Deleted = 0;

        cout<<"\n\nEnter the Name to be deleted : ";
        gets(sn);

        fstream f1,f2;
        f1.open("Tel.dat",ios::binary|ios::in);
        f2.open("TEMP.DAT",   ios::binary|ios::out);

        while( f1.read((char*) &P, sizeof(P)) )
        {
                if(P.CheckN(sn)==1)
                        Deleted ++ ;
                          else
                        f2.write((char *) &P, sizeof(P));
        }

        f1.close();
        f2.close();



        if (Deleted == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                {
                        remove("Tel.dat");
                        rename("TEMP.DAT" , "Tel.dat");
                }

}



void main()
{
        char choice;
        do{
```

```
        clrscr();
        cout<<"Menu\n";
        cout<<"Press 1 for APPEND A RECORD \n";
        cout<<"Press 2 for DISPLAY ALL RECORDS \n";
        cout<<"Press 3 for SEARCH BY NAME \n";
        cout<<"Press 4 for SEARCH BY TELEPHONE NUMBER \n";
        cout<<"Press 5 for SEARCH BY EMAIL ID \n";
        cout<<"Press 6 for MODIFY A RECORD \n";
        cout<<"Press 7 for DELETE A RECORD \n";
        cout<<"Press 8 to QUIT \n";
        cout<<"Enter your choice : ";
        choice=getche();
                switch(choice)
        {
            case '1':       Append();
                            break;
            case '2':       DisplayAll();
                            break;
            case '3':       SearchN();
                            break;
            case '4':       SearchT();
                            break;
            case '5':       SearchE();
                            break;
            case '6':       Modify();
                            break;
            case '7':       Delete();
                            break;
            case '8':               break;
            default :       cout<<"\nInvalid Choice Entered !!\n\n";
        }
        cout<<"\n\nPress any key to continue!";
        getch();
    }while(choice!='8');
}
```



```
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for SEARCH BY TELEPHONE NUMBER
Press 5 for SEARCH BY EMAIL ID
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 to QUIT
Enter your choice : 1
Enter the Detail of the Record :


Enter Name : Vikram
Enter Telephone number : 9881
Enter Email ID : vik@kic


Press any key to continue!_
```



```
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for SEARCH BY TELEPHONE NUMBER
Press 5 for SEARCH BY EMAIL ID
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 to QUIT
Enter your choice : 4

Enter the Telephone Number to be searched : 9881


Name : Vikram
Telephone number : 9881
Email ID : vik@kic



 TOTAL 1 RECORD(S) FOUND !!



Press any key to continue!
```

//39
/*WA menu based OOP to SEARCH, DISPLAY, ADD, DELETE, MODIFY and PROMOTE some records in a binary file STUDENT.DAT.
A record consists of name (20 char.), Class (int) and marks in five subjects (all int).
PROMOTE is a process to do the following:
Promote (increase the Class by 1) all those students whose total marks is greater than or equal to 300.
Delete all students who are in Class more than 12 (after promotion).
***/**

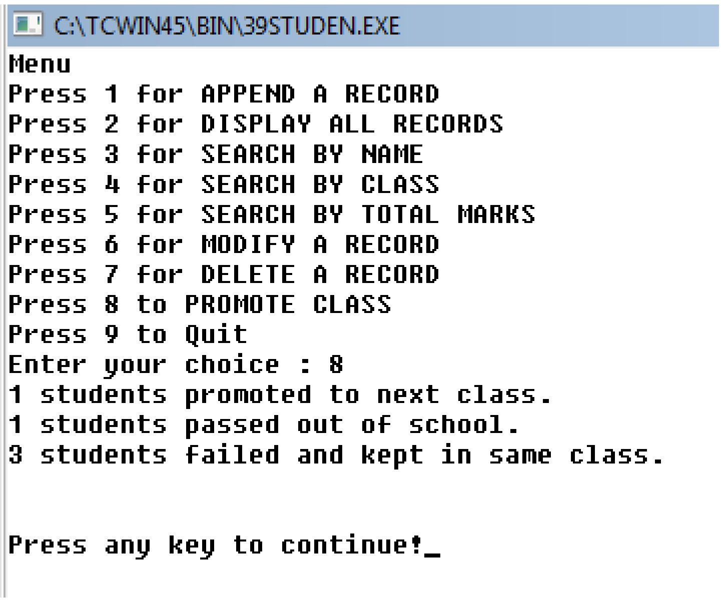
```
#include<fstream.h>
#include<string.h>
#include<stdio.h>
#include<conio.h>


class Student
{
                char Name[20];
                int Class;
                int Marks[5];
                int TotalMarks()
                {
                        int tm=0;
                        for(int i=0;i<5;i++)
                        tm+=Marks[i];
                        return tm;
                }

        public:
                void Getdata();
                void Showdata();
                int  CheckN(char *);
                int  CheckC(int);
                int  CheckTM(int);
```

```
		void Edit();
		int Promote();
    void ClassInc();
};



//	**************************** INPUT AN OBJECT *****************************
void Student::Getdata()
{
	cout<<"\n\nEnter Name : ";
	gets(Name);
	cout<<"\nEnter Class : ";
	cin>>Class;
	for(int i=0;i<5;i++)
	{
		cout<<"\nEnter Marks for subject"<<(i+1)<<"  : ";
		cin>>Marks[i];
	}
}



//	**************************** DISPLAY AN OBJECT *****************************
void Student::Showdata()
{
	cout<<"\n\nName : "<<Name<<endl;
	cout<<"Class: "<<Class<<endl;
	for(int i=0;i<5;i++)
	{
		cout<<"Marks for subject "<<(i+1)<<" : ";
		cout<<Marks[i]<<endl;
	}
}
//	**************************** CHECK NAME FOR AN OBJECT *****************************
int Student::CheckN(char name[])
{

	if (strcmp(name,Name)==0)
		return 1;
	else
		return 0;
}


//	**************************** CHECK CLASS FOR AN OBJECT *****************************
int Student::CheckC(int classs)
{
	if (classs==Class)
		return 1;
	else
```

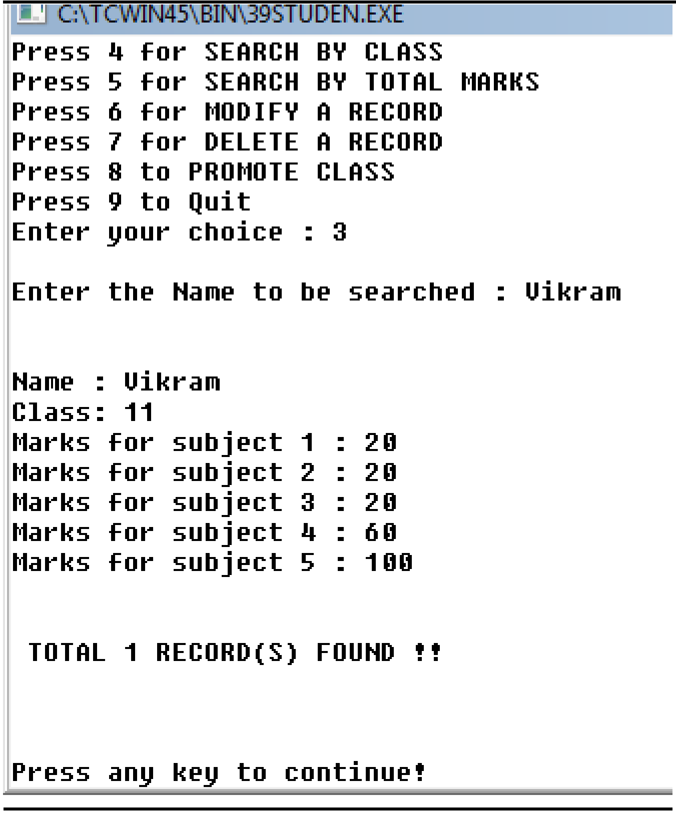
```
                return 0;
}




//      ****************** CHECK TOTAL MARKS FOR AN OBJECT **********************
int Student::CheckTM(int tm)
{
        if (tm==TotalMarks())
                return 1;
        else
                return 0;
}



//      *************************** EDIT/ MODIFY AN OBJECT ****************************
void Student::Edit()
{
        char choice;
  int i;
        do{
                cout<<"Current detail of the Record :\n\n";
                Showdata();
                cout<<"\n\nPress A if you want to change Name \n";
                cout<<"Press B if you want to change Class \n";
                cout<<"Press C if you want to change Marks  \n";
                cout<<"Press D if you want no more change \n";
                cout<<"Enter your option :";
                choice=getche();
                switch(choice)
                {
                        case 'A':
                        case 'a':       cout<<"\nEnter new Name : ";
                                                gets(Name);
                                                break;
                        case 'B':
                        case 'b':       cout<<"\nEnter new Class: ";
                                                cin>>Class;
                                                break;
                        case 'C':
                        case 'c':       cout<<"\nEnter new Marks  : ";
                                                for(i=0;i<5;i++)
                                                {
                                                        cout<<"\nEnter Marks for
subject"<<(i+1)<<" : ";
                                                        cin>>Marks[i];
                                                }
                                                break;
                        case 'D':
                        case 'd':       break;

                        default :       cout<<"\nInvalid Choice !! Try Again !!";
                }
        }while(choice!='D'&&choice!='d');
}

//****************Promote a student to the next class************************
int Student::Promote()
{
        int flag=0;
        if(TotalMarks()>=300)
        {
                if(Class==12)
                        flag=1;
                else
                        flag=2;
        }
  return flag;
}
//*********CLASS INCREMENT*********
void Student::ClassInc()
{
        Class++;
}

//      *************************** APPEND (ADD) A NEW RECORD ****************************
void Append()
{

        Student S;
        cout<<"\nEnter the Detail of the Record : \n";
        S.Getdata();

        fstream f1;
        f1.open("STUDENT.DAT",ios::binary|ios::app);

        f1.write((char*)&S,sizeof(S));

        f1.close();

}




//      *************************** DISPLAY ALL RECORDS ****************************
void DisplayAll()
{
        Student S;

        fstream f1;
        f1.open("STUDENT.DAT",ios::binary|ios::in);

        while(f1.read((char*)&S,sizeof(S)))
                S.Showdata();

        f1.close();

}


//      *************************** SEARCH A RECORD BY NAME ****************************
void SearchN()
{

        char sn[50];
        Student S;
        int Found = 0;

        cout<<"\n\nEnter the Name to be searched : ";
        gets(sn);

        fstream f1;
        f1.open("STUDENT.DAT",ios::binary|ios::in);

        while(f1.read((char*)&S,sizeof(S)))
        {
                if(S.CheckN(sn)==1)
                {
                        Found++ ;
                        S.Showdata();

                }
        }

        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<<Found<< " RECORD(S) FOUND !! \n\n";
}
```

```
//	**************************** SEARCH A RECORD BY CLASS*****************************
void SearchC()
{

	int sC;
	Student S;
	int Found = 0;

	cout<<"\n\nEnter the Class to be searched : ";
	cin>>sC;

	fstream f1;
	f1.open("STUDENT.DAT",ios::binary|ios::in);

	while( f1.read((char*) &S, sizeof(S)) )
	{
		if(S.CheckC(sC)==1)
		{
			Found++ ;
			S.Showdata();

		}
	}

	f1.close();

	if (Found == 0)
		cout<<"\n\n NO MATCH FOUND !! \n\n";
		  else
		cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}







//	**************************** SEARCH A RECORD BY TOTAL MARKS
*****************************
void SearchTM()
{

	int sTM;
	Student S;
	int Found = 0;

	cout<<"\n\nEnter the Total Marks  to be searched : ";
	cin>>sTM;
```

```
        fstream f1;
        f1.open("STUDENT.DAT",ios::binary|ios::in);

        while( f1.read((char*) &S, sizeof(S)) )
        {
                if(S.CheckTM(sTM)==1)
                {
                        Found++ ;
                        S.Showdata();

                }
        }

        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";

}





//      **************************** MODIFY A RECORD *****************************
void Modify()
{

        char sn[50];
        Student S;
        int Modified = 0;

        cout<<"\n\nEnter the Name to be modified : ";
        gets(sn);

        fstream f1("STUDENT.DAT",ios::binary|ios::in|ios::out);


        while(f1.read((char*)&S,sizeof(S)) )
        {
                if(S.CheckN(sn)==1)
                {
                        Modified++ ;
                        S.Edit();
                }
                f1.write((char*)&S,sizeof(S));
        }
        f1.close();


        if (Modified==0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
}




//      ************************** DELETE A RECORD ***************************
void Delete()
{
        char sn[50];
        Student S;
        int Deleted = 0;

        cout<<"\n\nEnter the Name to be deleted : ";
        gets(sn);

        fstream f1,f2;
        f1.open("STUDENT.DAT",ios::binary|ios::in);
        f2.open("TEMP.DAT",   ios::binary|ios::out);

        while( f1.read((char*) &S, sizeof(S)) )
        {
                if(S.CheckN(sn)==1)
                        Deleted ++ ;
                          else
                        f2.write((char*)&S,sizeof(S));
        }

        f1.close();
        f2.close();



        if (Deleted == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                {
                        remove("STUDENT.DAT");
                        rename("TEMP.DAT" , "STUDENT.DAT");
                }

}
//****************Promote the Class*************************
void PromoteAll()
{
        Student S;
        int Promoted=0;
        int Failed=0;
        int PassedOut=0;

        fstream f1,f2;
        f1.open("STUDENT.DAT",ios::binary|ios::in);
        f2.open("TEMP.DAT",   ios::binary|ios::out);

        while(f1.read((char*)&S,sizeof(S)))
        {
                if(S.Promote()==0)
                {
                        f2.write((char*)&S,sizeof(S));
                        Failed++;
                }
                else if(S.Promote()==2)
                {
                        S.ClassInc();
                        Promoted++;
                        f2.write((char*)&S,sizeof(S));
                }
                else
                {

                        PassedOut++;
                }
        }
        cout<<endl<<Promoted<<" students promoted to next class.\n";
        cout<<PassedOut<<" students passed out of school.\n";
        cout<<Failed<<" students failed and kept in same class.\n";

  remove("STUDENT.DAT");
        rename("TEMP.DAT" , "STUDENT.DAT");
}


void main()
{
        char choice;
        do{
                clrscr();
                cout<<"Menu\n";
                cout<<"Press 1 for APPEND A RECORD \n";
                cout<<"Press 2 for DISPLAY ALL RECORDS \n";
                cout<<"Press 3 for SEARCH BY NAME \n";
                cout<<"Press 4 for SEARCH BY CLASS\n";
                cout<<"Press 5 for SEARCH BY TOTAL MARKS  \n";
                cout<<"Press 6 for MODIFY A RECORD \n";
                cout<<"Press 7 for DELETE A RECORD \n";
                cout<<"Press 8 to PROMOTE CLASS \n";
                cout<<"Press 9 to Quit \n";
                cout<<"Enter your choice : ";
                choice=getche();
		switch(choice)
		{
			case '1':	Append();
						break;
			case '2':	DisplayAll();
						break;
			case '3':	SearchN();
						break;
			case '4':	SearchC();
						break;
			case '5':	SearchTM();
						break;
			case '6':	Modify();
						break;
			case '7':	Delete();
						break;
			case '8':	PromoteAll();
						break;
			case '9':
						break;
			default :cout<<"\nInvalid Choice Entered !!\n\n";
		}
		cout<<"\n\nPress any key to continue!";
		getch();
	}while(choice!='9');
}
```



```
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for SEARCH BY CLASS
Press 5 for SEARCH BY TOTAL MARKS
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 to PROMOTE CLASS
Press 9 to Quit
Enter your choice : 8
1 students promoted to next class.
1 students passed out of school.
3 students failed and kept in same class.


Press any key to continue!_
```



```
Press 4 for SEARCH BY CLASS
Press 5 for SEARCH BY TOTAL MARKS
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 to PROMOTE CLASS
Press 9 to Quit
Enter your choice : 3

Enter the Name to be searched : Vikram


Name : Vikram
Class: 11
Marks for subject 1 : 20
Marks for subject 2 : 20
Marks for subject 3 : 20
Marks for subject 4 : 60
Marks for subject 5 : 100


 TOTAL 1 RECORD(S) FOUND !!



Press any key to continue!
```

//42
/*WA menu based OOP to ADD, DELETE and DISPLAY of some records stored in a binary file HALLFAME.DAT.
A record consists of name (20 characters) and points (integer).
The addition and deletion of records are done in such way that the records are always stored in descending order of their points.
***/**

```
#include<fstream.h>
#include<string.h>
#include<stdio.h>

#include<conio.h>


class Score
{
		char Name[50];
		int score;
	public:
		void Getdata();
		void Showdata();
		int  CheckN(char *);
		int ScrCmp(Score);
};



//	**************************** INPUT AN OBJECT *****************************
void Score::Getdata()
{
	cout<<"\n\nEnter Name : ";
	gets(Name);
	cout<<"\nEnter Score: ";
	cin>>score;

}




//	**************************** DISPLAY AN OBJECT ****************************
void Score::Showdata()
{
	cout<<"\n\nName : "<<Name<<endl;
	cout<<"Score: "<<score<<endl;

}
//	**************************** CHECK NAME FOR AN OBJECT ****************************
int Score::CheckN(char name[])
{
        if (strcmp(name,Name)==0)
                return 1;
        else
                return 0;
}




//      ******************** COMPARE SCORES OF OBJECTS ********************
int Score::ScrCmp(Score A)
{
        if (score<A.score)
        {
                return 1;
        }
        else
                return 0;
}


//      *************************** APPEND (ADD) A NEW RECORD ****************************
void Append()
{

        Score S,P;
        cout<<"\nEnter the Detail of the Record : \n";
        S.Getdata();

        int flag=0;

        fstream f1,f2;
        f1.open("HALLFAME.DAT",ios::binary|ios::in);
        f2.open("TEMP.DAT",ios::binary|ios::out);


        while(f1.read((char*)&P,sizeof(P))&&flag==0)
        {
                if((S.ScrCmp(P)==1)
                        f2.write((char*)&P,sizeof(P));
                else
                {
                        f2.write((char*)&S,sizeof(S));
                        f2.write((char*)&P,sizeof(P));
                        flag=1;
                }
        }
        while(f1.read((char*)&P,sizeof(P)))
        f2.write((char*)&P,sizeof(P));
        f1.close();
        f2.close();

        remove("HALLFAME.DAT");
        rename("TEMP.DAT","HALLFAME.DAT");

}




//      **************************** DISPLAY ALL RECORDS *****************************
void DisplayAll()
{

        Score S;

        fstream f1;
        f1.open("HALLFAME.DAT",ios::binary|ios::in);
  cout<<"\nHall Of Fame:\n";
        while(f1.read((char*)&S,sizeof(S)))
                S.Showdata();

        f1.close();

}


//      **************************** SEARCH A RECORD BY NAME *****************************
void SearchN()
{

        char sn[50];
        Score S;
        int Found = 0;

        cout<<"\n\nEnter the Name to be searched : ";
        gets(sn);

        fstream f1;
        f1.open("HALLFAME.DAT",ios::binary|ios::in);

        while(f1.read((char*)&S,sizeof(S)))
        {
                if(S.CheckN(sn)==1)
                {
                        Found++ ;
                        S.Showdata();

                }
        }
        f1.close();

        if (Found == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                cout<<"\n\n TOTAL "<<Found<< " RECORD(S) FOUND !! \n\n";
}



//      *************************** DELETE A RECORD ****************************
void Delete()
{
        char sn[50];
        Score S;
        int Deleted = 0;

        cout<<"\n\nEnter the Name to be deleted : ";
        gets(sn);

        fstream f1,f2;
        f1.open("HALLFAME.DAT",ios::binary|ios::in);
        f2.open("TEMP.DAT",   ios::binary|ios::out);

        while( f1.read((char*) &S, sizeof(S)) )
        {
                if(S.CheckN(sn)==1)
                        Deleted ++ ;
                          else
                        f2.write((char *) &S, sizeof(S));
        }

        f1.close();
        f2.close();



        if (Deleted == 0)
                cout<<"\n\n NO MATCH FOUND !! \n\n";
                  else
                {
                        remove("HALLFAME.DAT");
                        rename("TEMP.DAT" , "HALLFAME.DAT");
                }

}



void main()
{
        char choice;
        do{
                clrscr();
                cout<<"Menu\n";
                cout<<"Press 1 for APPEND A RECORD \n";
                cout<<"Press 2 for DISPLAY ALL RECORDS \n";
                cout<<"Press 3 for SEARCH BY NAME \n";
                cout<<"Press 4 for DELETE A RECORD \n";
                cout<<"Press 5 to QUIT \n";
                cout<<"Enter your choice : ";
                choice=getche();
                switch(choice)
                {
                        case '1':       Append();
                                        break;
                        case '2':       DisplayAll();
                                        break;
                        case '3':       SearchN();
                                        break;
                        case '4':       Delete();
                                        break;
                        case '5':
                                        break;
                        default :       cout<<"\nInvalid Choice Entered !!\n\n";
                }
                cout<<"\n\nPress any key to continue!";
                getch();
        }while(choice!='5');
}
```

//Game (Use with Hall of Fame)

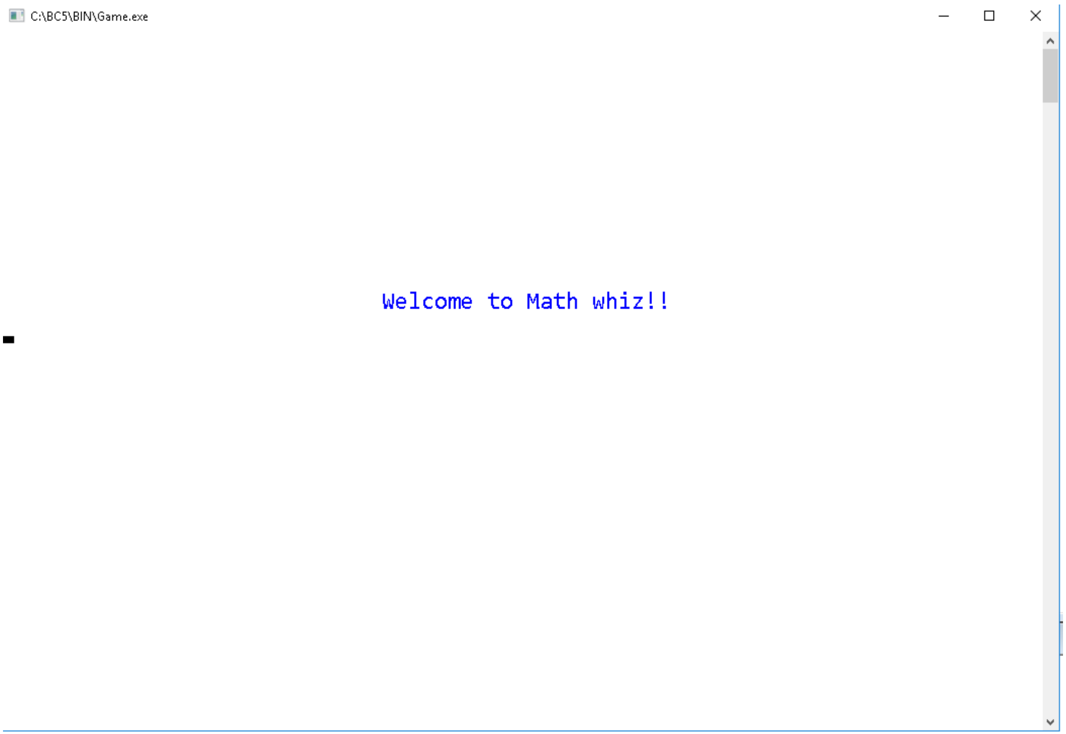
```
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>

int p,q,r,s;

void main()
{
    randomize();

   getch();
   gotoxy(30,10);
   cout<<"Welcome to Math whiz!!"<<endl;  //Game name

   for(int i=0;i<1000;i++)
   for(int i=0;i<1000;i++)
   for(int i=0;i<4;i++)
   for(int i=0;i<1000;i++);    //Delay using empty loop

   clrscr();

   gotoxy(28,10);
   cout<<"A Fun and addictive math game"<<endl;
   for(int i=0;i<1000;i++)
   for(int i=0;i<1000;i++)
   for(int i=0;i<4;i++)
   for(int i=0;i<1000;i++);

   clrscr();

   gotoxy(27,10);
   cout<<"Testing your skills is finally fun"<<endl;
   for(int i=0;i<1000;i++)
   for(int i=0;i<1000;i++)
   for(int i=0;i<4;i++)
   for(int i=0;i<1000;i++);
   clrscr();



   gotoxy(25,10);
   cout<<"Enter Difficulty level";
   cout<<"\n 1. Easy \n 2. Medium \n 3. Hard";       //Input Difficulty level
   gotoxy(25,14);
   cout<<"Enter Corresponding number";
   gotoxy(35,15);
   char dl;
```

```
  cin>>dl;







      switch(dl)
      {
           case'1':              clrscr();
                                 gotoxy(25,10);
                                 cout<<"Difficutly Level Chosen: Easy";
                                 for(p=0;p<1000;p++)
                                 for(q=0;q<1000;q++)
                                 for(r=0;r<10;r++)
                                 for(s=0;s<100;s++);
                                 clrscr();
                                 break;
           case'2':              clrscr();
                                 gotoxy(25,10);
                                 cout<<"Difficutly Level Chosen: Medium";
                                 for(p=0;p<1000;p++)
                                 for(q=0;q<1000;q++)
                                 for(r=0;r<10;r++)
                                 for(s=0;s<100;s++);
                                 clrscr();
                                 break;
           case'3':              clrscr();
                                 gotoxy(25,10);
                                 cout<<"Difficutly Level Chosen: Hard";
                                 for(p=0;p<1000;p++)
                                 for(q=0;q<1000;q++)
                                 for(r=0;r<10;r++)
                                 for(s=0;s<100;s++);
                                 clrscr();
                                 break;

           default:              gotoxy(30,16);
                                 cout<<"Invalid Level";
      }
```

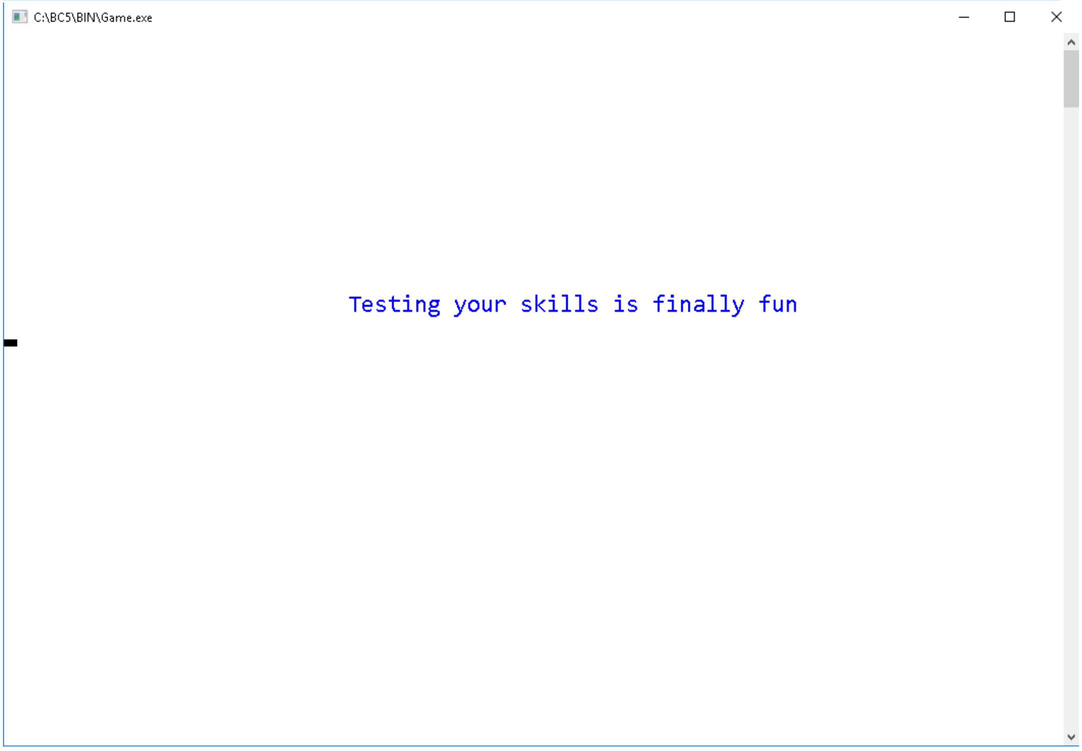
```
if(dl=='1')
 {
    int count=0,score=0;
    while(count<3)
    {
        double ans,num1,num2;

        num1=10+random(90)+0.1*(random(10));//Random number
        num2=10+random(90)+0.1*(random(10));//with one decimal
        gotoxy(30,10);
        cout<<"Calculate The Sum";
        gotoxy(31,11);
        cout<<num1;
        gotoxy(31,12);
        cout<<num2;
        gotoxy(31,13);
        cin>>ans;
        if(ans==num1+num2)
        {
            gotoxy(31,14);
            cout<<"Correct";
            score+=10;          //Points tally
        }
        else
        {
            gotoxy(31,14);
            cout<<"Wrong Ans";
        }
        for(p=0;p<1000;p++)
        for(q=0;q<1000;q++)
        for(r=0;r<8;r++)
        for(s=0;s<100;s++);
        clrscr();
        count++;
   }
   gotoxy(30,10);
   cout<<"Game over";
   gotoxy(30,11);
   cout<<"Your score"<<score;
}
```

```
if(dl=='2')
  {
        int count=0,score=0;
        while(count<3)
        {
                double ans,num1,num2;

                num1=100+random(900)+0.1*(random(10))+0.01*(random(10));
                num2=100+random(900)+0.1*(random(10))+0.01*(random(10));
                gotoxy(30,10);
                cout<<"Calculate The Sum";
                gotoxy(31,11);
                cout<<num1;
                gotoxy(31,12);
                cout<<num2;
                gotoxy(31,13);
                cin>>ans;
                if(ans==num1+num2)
                {
                        gotoxy(31,14);
                        cout<<"Correct";
                        score+=20;
                }
                else
                {
                        gotoxy(31,14);
                        cout<<"Wrong Ans";
                }
                for(p=0;p<1000;p++)
                for(q=0;q<1000;q++)
                for(r=0;r<8;r++)
                for(s=0;s<100;s++);
                clrscr();
                count++;
        }
        gotoxy(30,10);
        cout<<"Game over";
        gotoxy(30,11);
        cout<<"Your score"<<score;
}
```

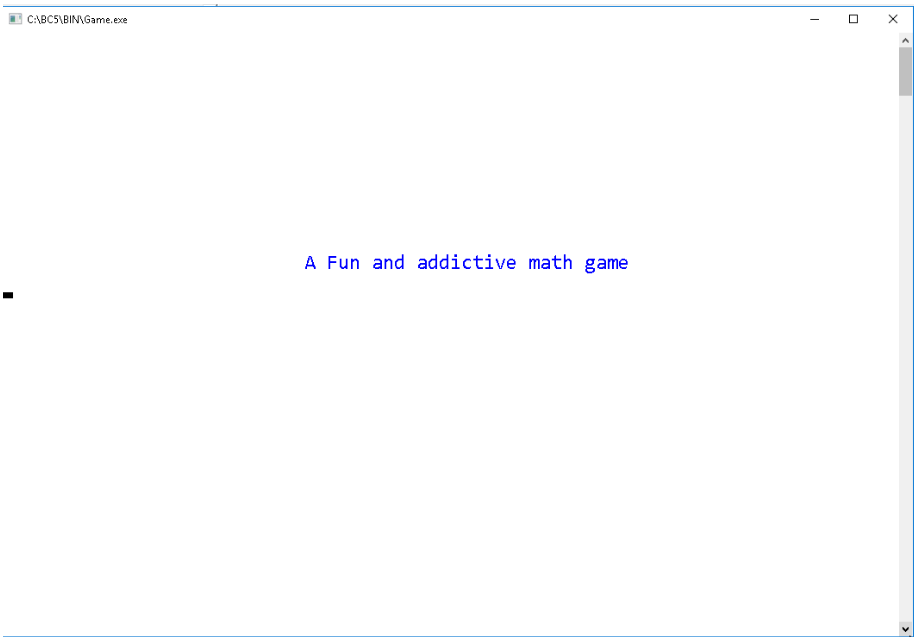
```
if(dl=='3')
  {
        int count=0,score=0;
        while(count<3)
        {
                double ans,num1,num2;

    num1=100+random(9000)+0.1*(random(10))+0.01*(random(10))+(0.001*random(10));
    num2=100+random(9000)+0.1*(random(10))+0.01*(random(10))+(0.001*random(10));
                gotoxy(30,10);
                cout<<"Calculate The Sum";
                gotoxy(31,11);
                cout<<num1;
                gotoxy(31,12);
                cout<<num2;
                gotoxy(31,13);
                cin>>ans;
                if(ans==num1+num2)
                {
                        gotoxy(31,14);
                        cout<<"Correct";
                        score+=30;
                }
                else
                {
                        gotoxy(31,14);
                        cout<<"Wrong Ans";
                }
                for(p=0;p<1000;p++)
                for(q=0;q<1000;q++)
                for(r=0;r<8;r++)
                for(s=0;s<100;s++);
                clrscr();
                count++;
        }
        gotoxy(30,10);
        cout<<"Game over";
        gotoxy(30,11);
        cout<<"Your score"<<score;
        }
gotoxy(30,12);
cout<<"Thanks for Playing";
getch();

}
```



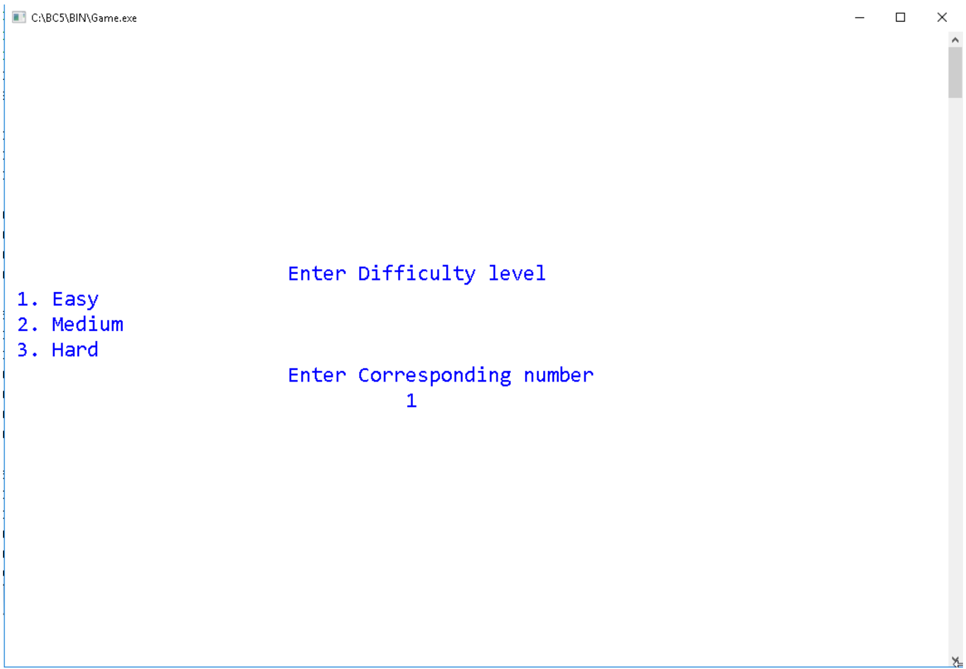
```
Welcome to Math whiz!!
```



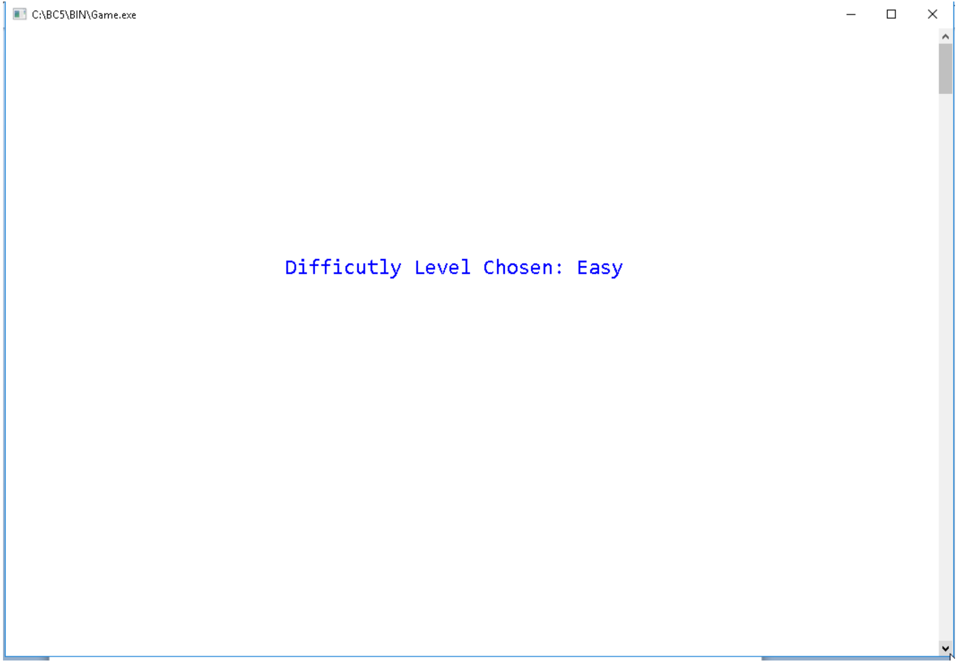
```
Testing your skills is finally fun
```



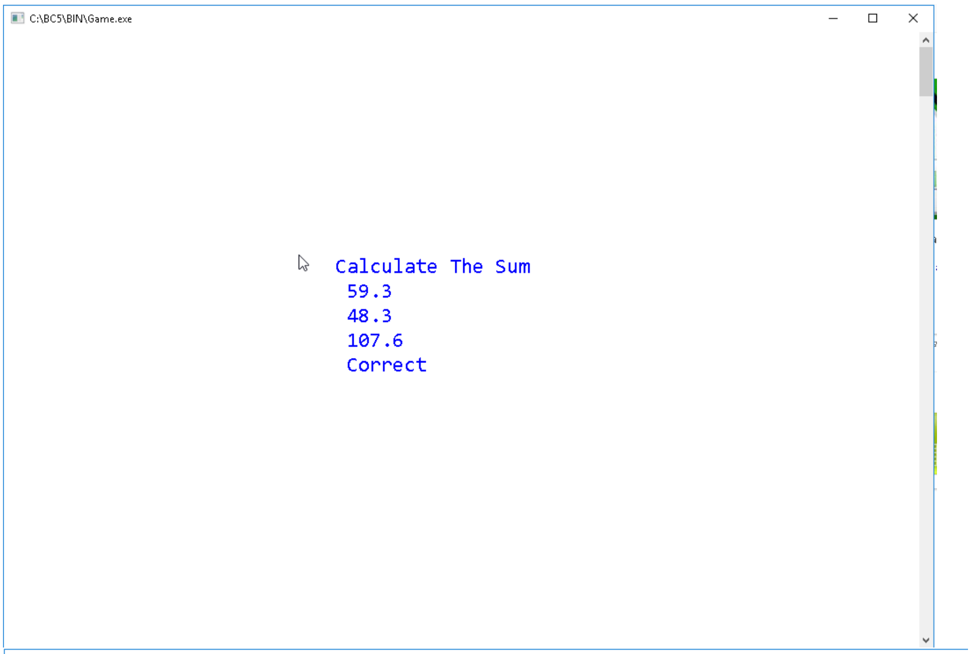
```
A Fun and addictive math game
```



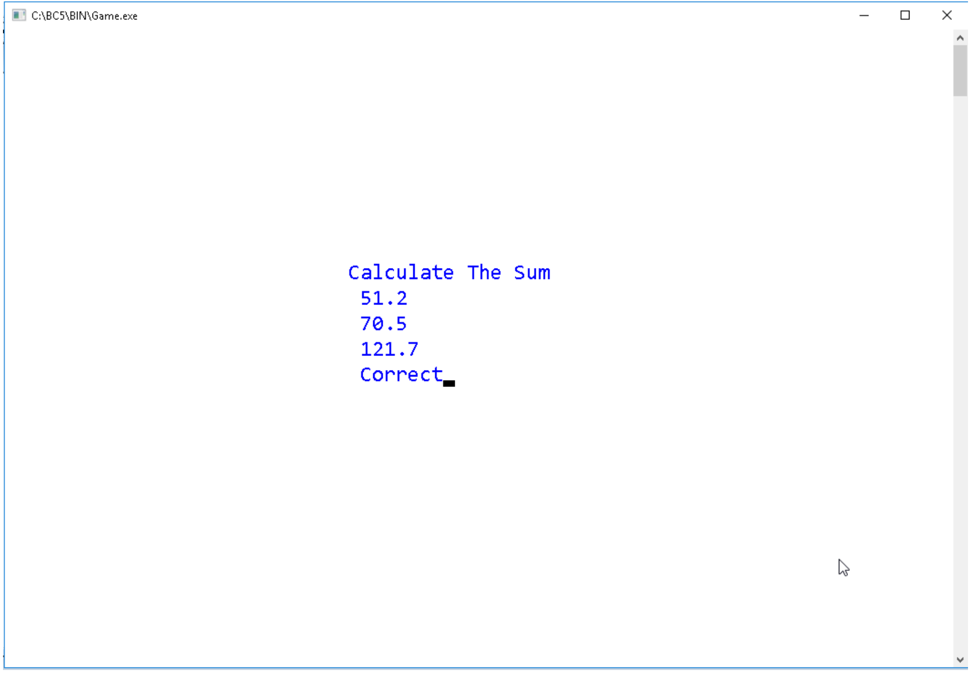
```
                                Enter Difficulty level
1. Easy
2. Medium
3. Hard

                                Enter Corresponding number
                                        1
```



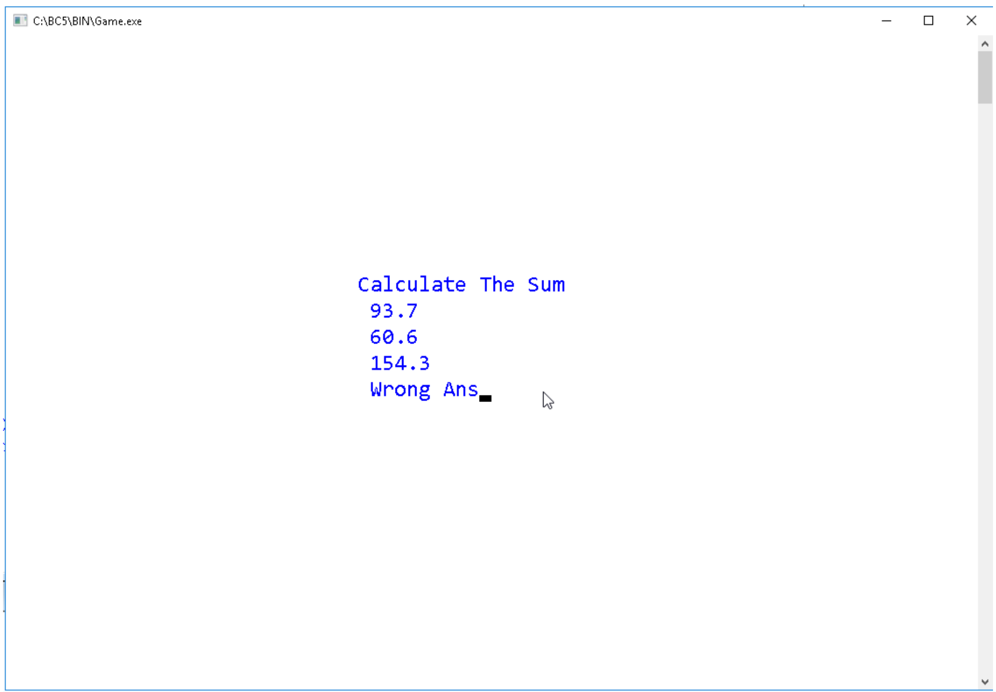
```
Difficutly Level Chosen: Easy
```



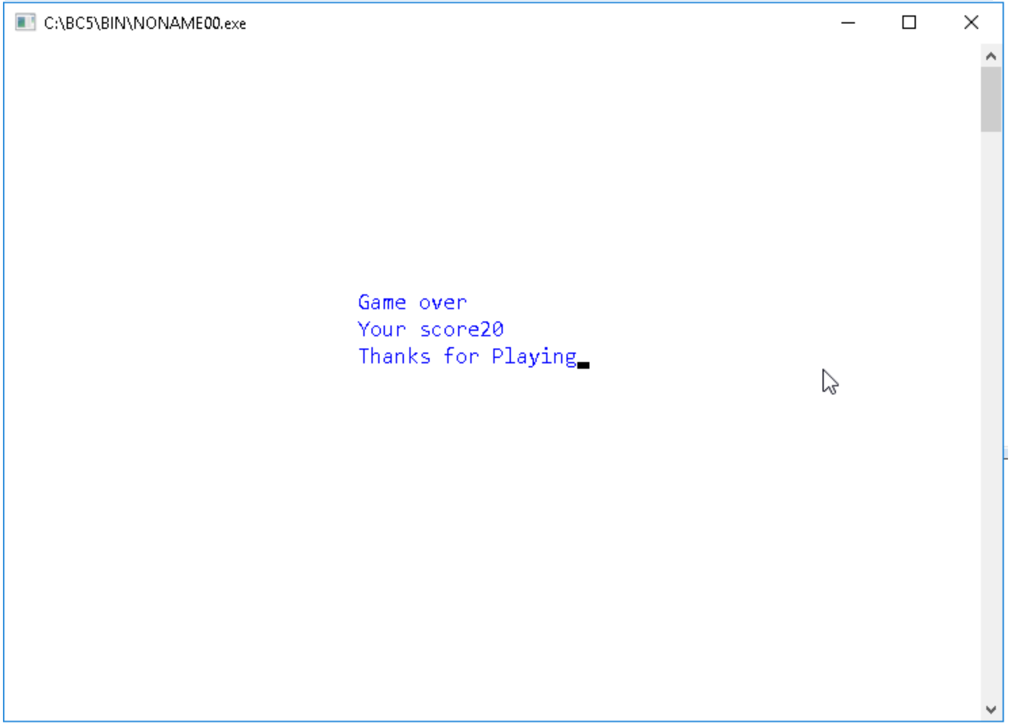
```
Calculate The Sum
 59.3
 48.3
 107.6
 Correct
```



```
Calculate The Sum
 51.2
 70.5
 121.7
 Correct
```



```
Calculate The Sum
 93.7
 60.6
 154.3
 Wrong Ans
```

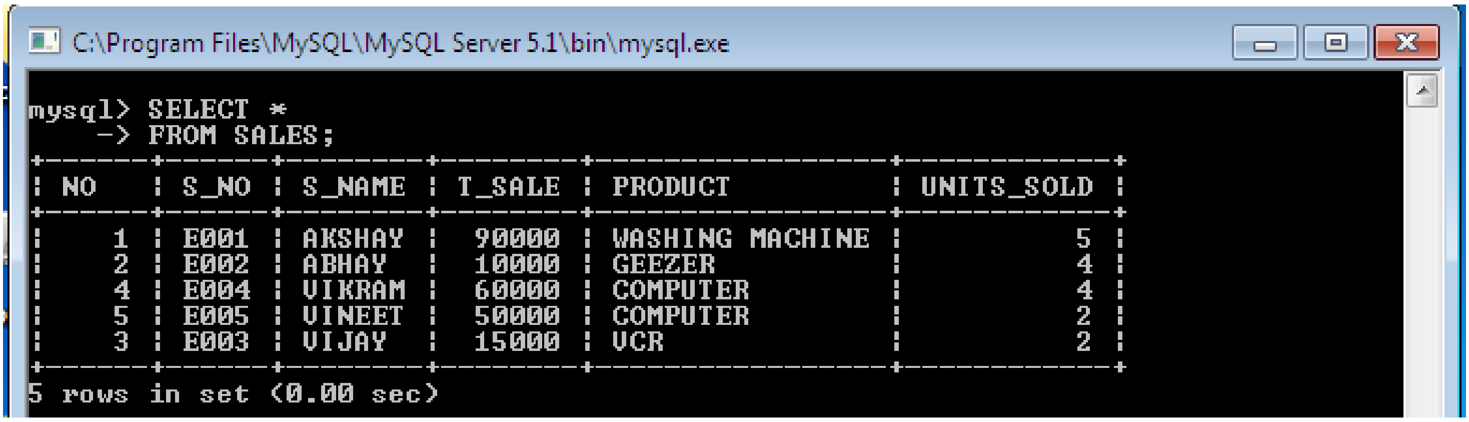


```
Game over
Your score20
Thanks for Playing
```

SQL QUERIES

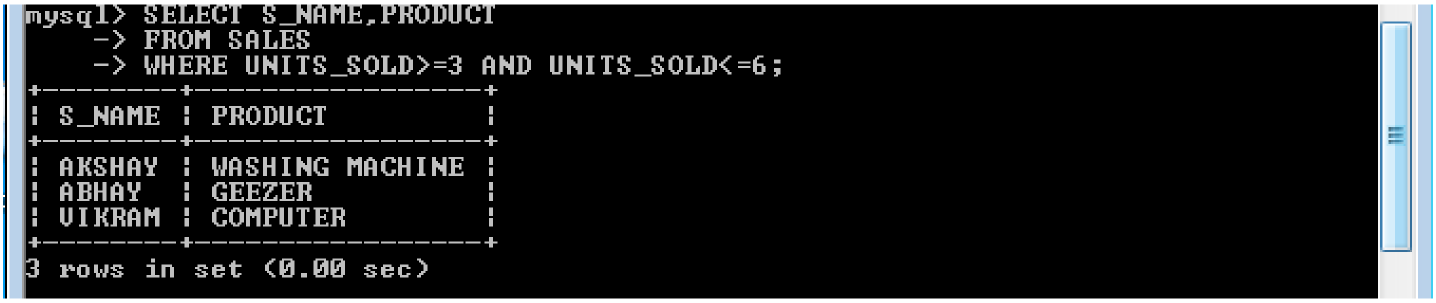
1) Create the following table in database “db”.Write the SQL commands for the following queries:

RELATION : SALES



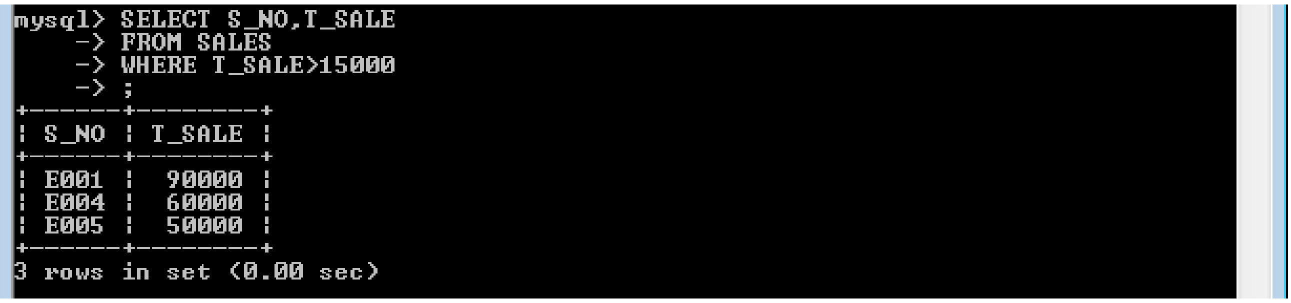
```
mysql> SELECT *
    -> FROM SALES;
+------+------+--------+--------+-----------------+------------+
| NO   | S_NO | S_NAME | T_SALE | PRODUCT         | UNITS_SOLD |
+------+------+--------+--------+-----------------+------------+
|    1 | E001 | AKSHAY |  90000 | WASHING MACHINE |          5 |
|    2 | E002 | ABHAY  |  10000 | GEEZER          |          4 |
|    4 | E004 | VIKRAM |  60000 | COMPUTER        |          4 |
|    5 | E005 | VINEET |  50000 | COMPUTER        |          2 |
|    3 | E003 | VIJAY  |  15000 | VCR             |          2 |
+------+------+--------+--------+-----------------+------------+
5 rows in set (0.00 sec)
```

a) To display the S_NaME and PRODucT where UNITS_SOLD are between 3 and 6.



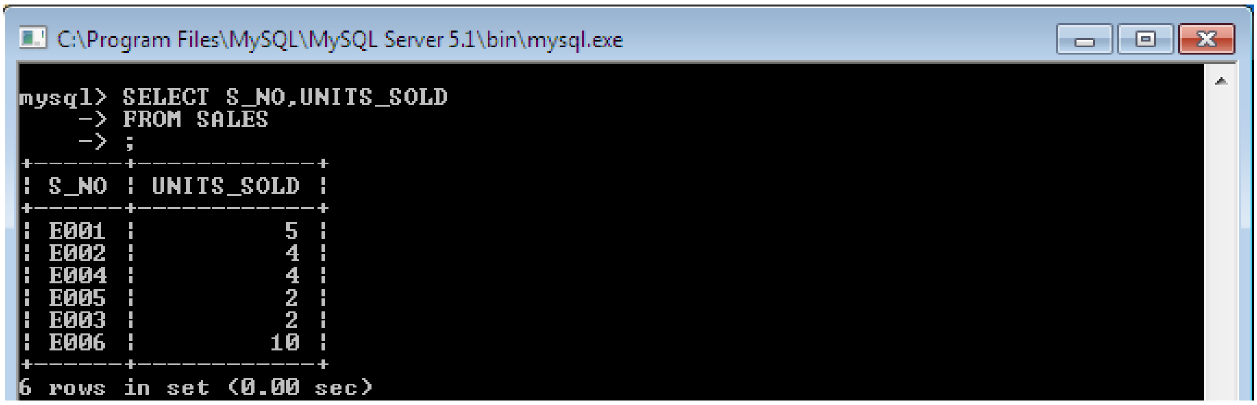
```
mysql> SELECT S_NAME,PRODUCT
    -> FROM SALES
    -> WHERE UNITS_SOLD>=3 AND UNITS_SOLD<=6;
+--------+-----------------+
| S_NAME | PRODUCT         |
+--------+-----------------+
| AKSHAY | WASHING MACHINE |
| ABHAY  | GEEZER          |
| VIKRAM | COMPUTER        |
+--------+-----------------+
3 rows in set (0.00 sec)
```

b) To display S_NO and T_SAE where T_SALE is greater than 15000



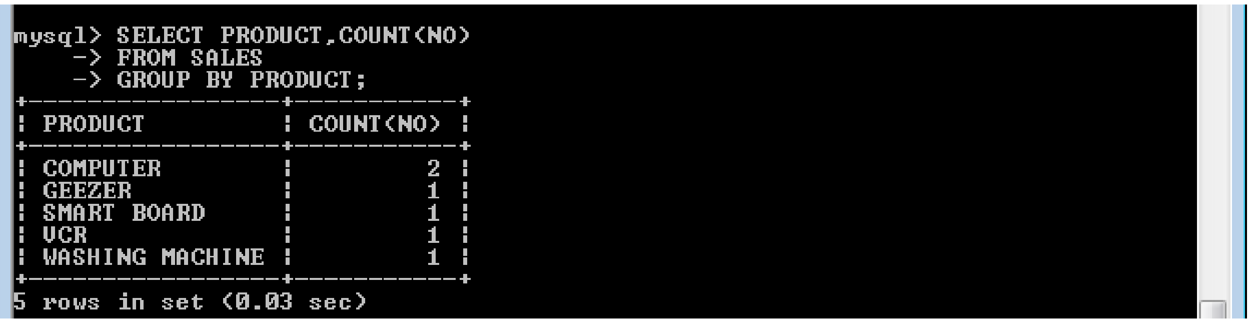
```
mysql> SELECT S_NO,T_SALE
    -> FROM SALES
    -> WHERE T_SALE>15000
    -> ;
+------+--------+
| S_NO | T_SALE |
+------+--------+
| E001 |  90000 |
| E004 |  60000 |
| E005 |  50000 |
+------+--------+
3 rows in set (0.00 sec)
```

c) To Display the S_NO and the no. of UNITS_SOLD of each salesman in the table SALES



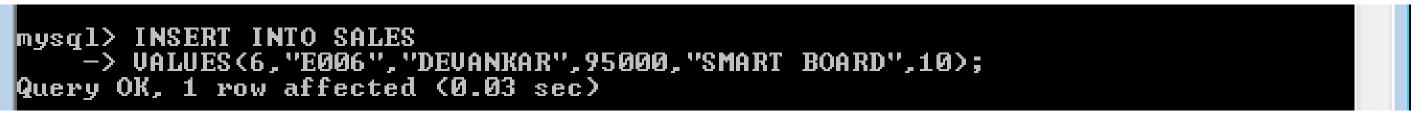
```
mysql> SELECT S_NO,UNITS_SOLD
    -> FROM SALES
    -> ;
+------+------------+
| S_NO | UNITS_SOLD |
+------+------------+
| E001 |          5 |
| E002 |          4 |
| E004 |          4 |
| E005 |          2 |
| E003 |          2 |
| E006 |         10 |
+------+------------+
6 rows in set (0.00 sec)
```

d) To display number of salesmen of each type of Products.



```
mysql> SELECT PRODUCT,COUNT(NO)
    -> FROM SALES
    -> GROUP BY PRODUCT;
+-----------------+-----------+
| PRODUCT         | COUNT(NO) |
+-----------------+-----------+
| COMPUTER        |         2 |
| GEEZER          |         1 |
| SMART BOARD     |         1 |
| VCR             |         1 |
| WASHING MACHINE |         1 |
+-----------------+-----------+
5 rows in set (0.03 sec)
```

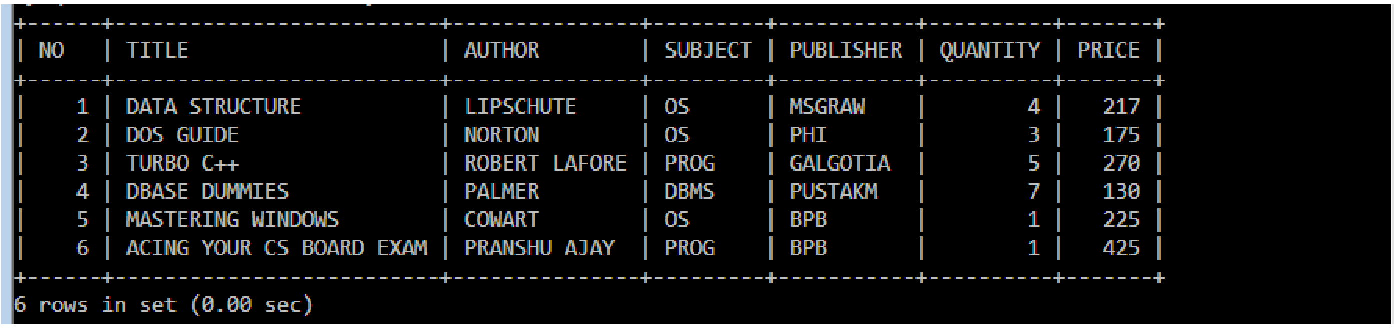
e) To insert a new row in the table with suitable values.



```
mysql> INSERT INTO SALES
    -> VALUES(6,"E006","DEVANKAR",95000,"SMART BOARD",10);
Query OK, 1 row affected (0.03 sec)
```

2) Create the following table in database “db”.Write the SQL commands for the following queries:

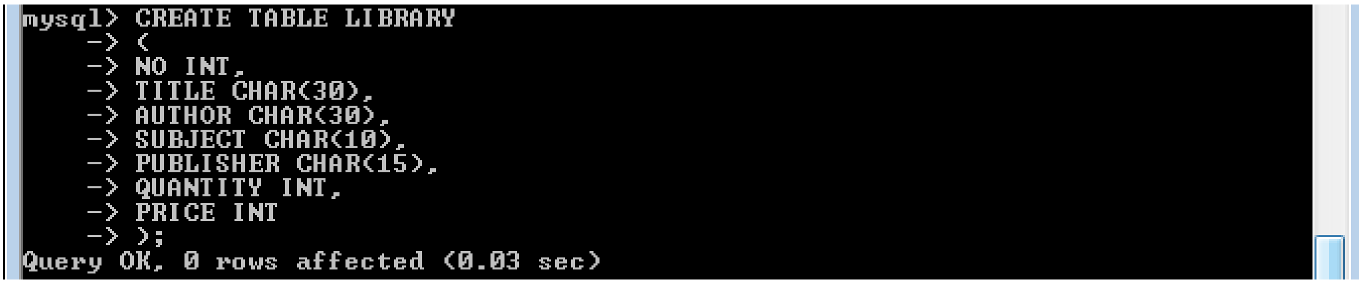
Relation: LIBRARY



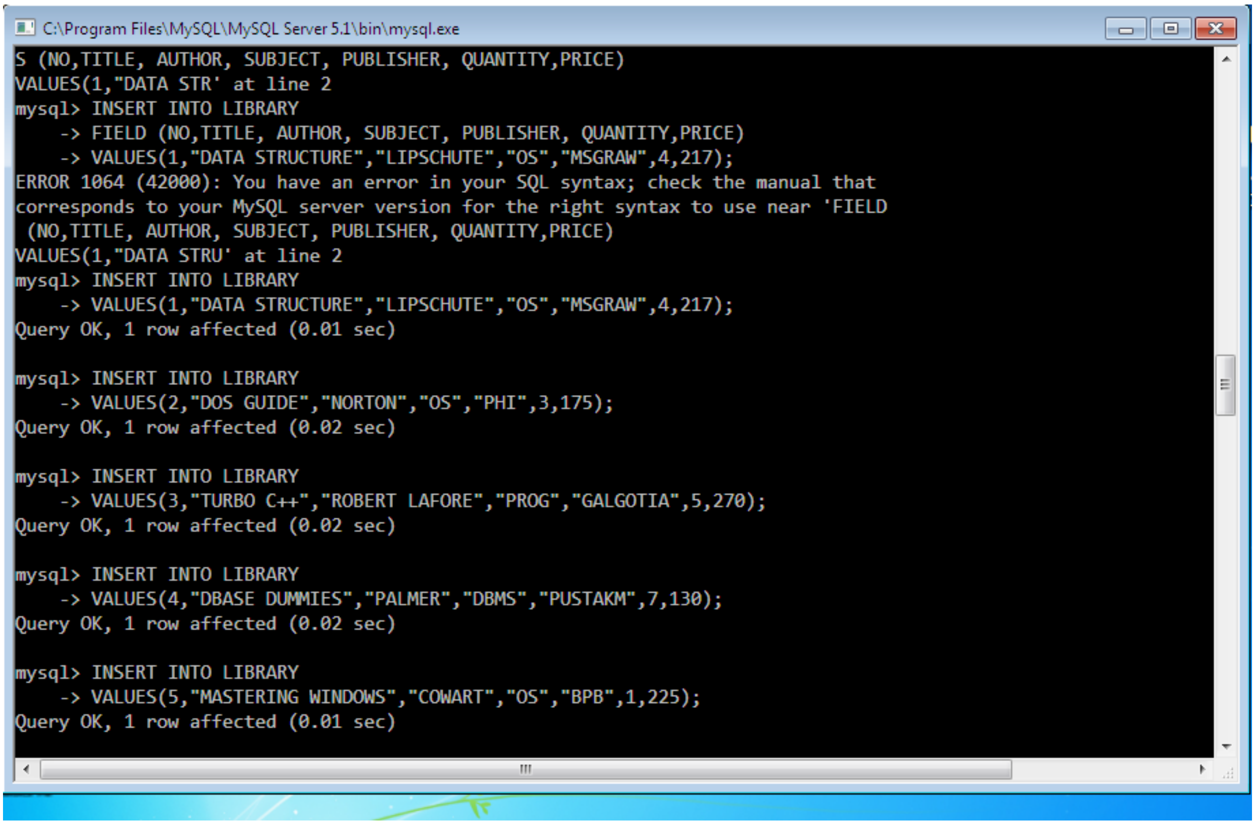
```
+------+-------------------------+---------------+---------+-----------+----------+-------+
| NO   | TITLE                   | AUTHOR        | SUBJECT | PUBLISHER | QUANTITY | PRICE |
+------+-------------------------+---------------+---------+-----------+----------+-------+
|    1 | DATA STRUCTURE          | LIPSCHUTE     | OS      | MSGRAW    |        4 |   217 |
|    2 | DOS GUIDE               | NORTON        | OS      | PHI       |        3 |   175 |
|    3 | TURBO C++               | ROBERT LAFORE | PROG    | GALGOTIA  |        5 |   270 |
|    4 | DBASE DUMMIES           | PALMER        | DBMS    | PUSTAKM   |        7 |   130 |
|    5 | MASTERING WINDOWS       | COWART        | OS      | BPB       |        1 |   225 |
|    6 | ACING YOUR CS BOARD EXAM | PRANSHU AJAY | PROG    | BPB       |        1 |   425 |
+------+-------------------------+---------------+---------+-----------+----------+-------+
6 rows in set (0.00 sec)
```

Ans:

->

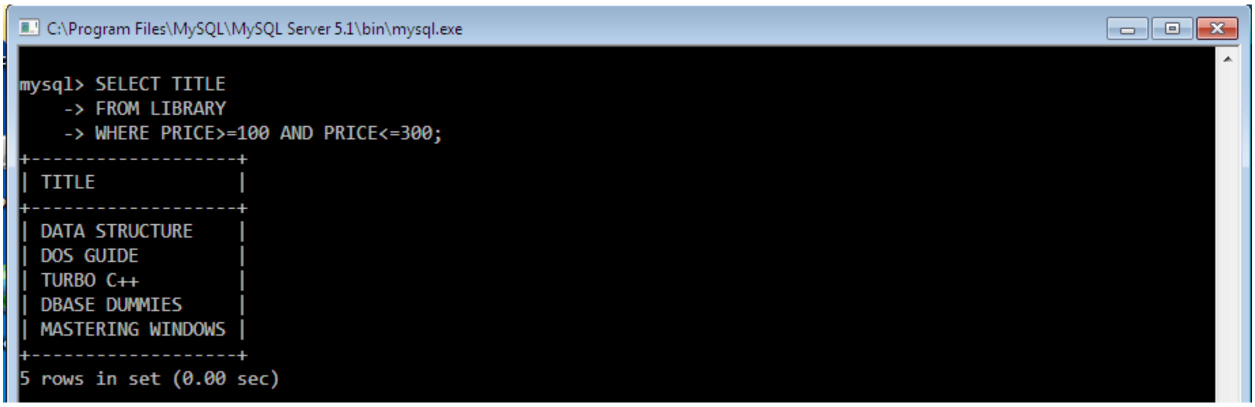


```
mysql> CREATE TABLE LIBRARY
    -> (
    -> NO INT,
    -> TITLE CHAR(30),
    -> AUTHOR CHAR(30),
    -> SUBJECT CHAR(10),
    -> PUBLISHER CHAR(15),
    -> QUANTITY INT,
    -> PRICE INT
    -> );
Query OK, 0 rows affected (0.03 sec)
```



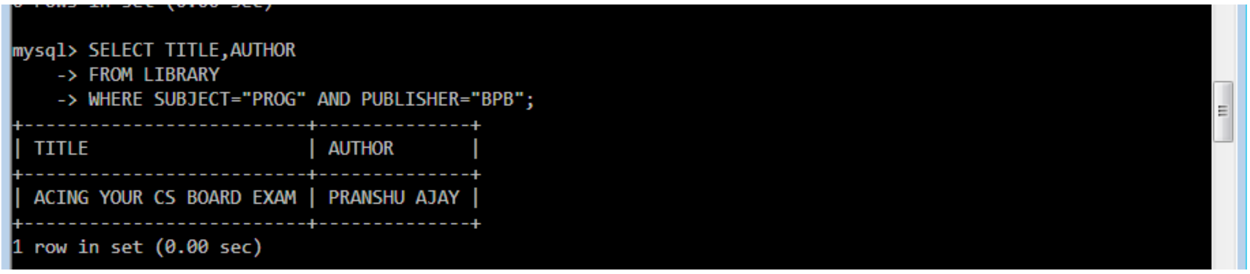
```
S (NO,TITLE, AUTHOR, SUBJECT, PUBLISHER, QUANTITY,PRICE)
VALUES(1,"DATA STR' at line 2
mysql> INSERT INTO LIBRARY
    -> FIELD (NO,TITLE, AUTHOR, SUBJECT, PUBLISHER, QUANTITY,PRICE)
    -> VALUES(1,"DATA STRUCTURE","LIPSCHUTE","OS","MSGRAW",4,217);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'FIELD
 (NO,TITLE, AUTHOR, SUBJECT, PUBLISHER, QUANTITY,PRICE)
VALUES(1,"DATA STRU' at line 2
mysql> INSERT INTO LIBRARY
    -> VALUES(1,"DATA STRUCTURE","LIPSCHUTE","OS","MSGRAW",4,217);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LIBRARY
    -> VALUES(2,"DOS GUIDE","NORTON","OS","PHI",3,175);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO LIBRARY
    -> VALUES(3,"TURBO C++","ROBERT LAFORE","PROG","GALGOTIA",5,270);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO LIBRARY
    -> VALUES(4,"DBASE DUMMIES","PALMER","DBMS","PUSTAKM",7,130);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO LIBRARY
    -> VALUES(5,"MASTERING WINDOWS","COWART","OS","BPB",1,225);
Query OK, 1 row affected (0.01 sec)
```

a) To display the title of all the books with Price between 100 and 300.



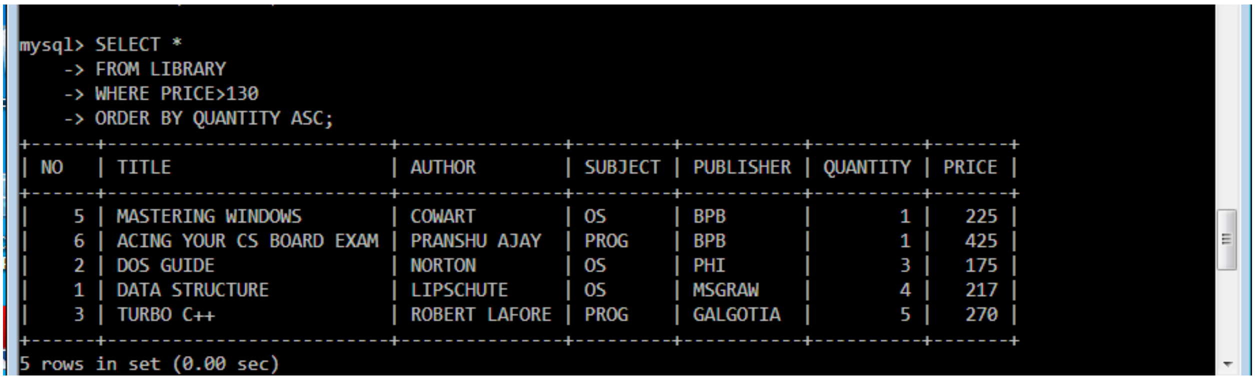
```
mysql> SELECT TITLE
    -> FROM LIBRARY
    -> WHERE PRICE>=100 AND PRICE<=300;
+-------------------+
| TITLE             |
+-------------------+
| DATA STRUCTURE    |
| DOS GUIDE         |
| TURBO C++         |
| DBASE DUMMIES     |
| MASTERING WINDOWS |
+-------------------+
5 rows in set (0.00 sec)
```

b) To display TITLE and AUTHOR of all books having Subject PROG and published by BPB.



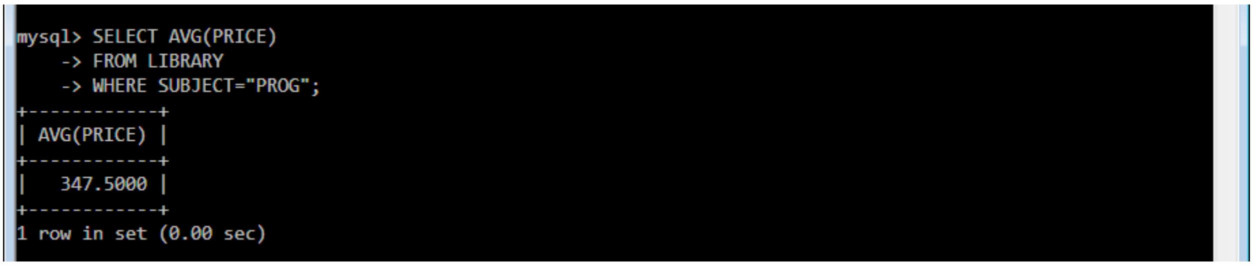
```
mysql> SELECT TITLE,AUTHOR
    -> FROM LIBRARY
    -> WHERE SUBJECT="PROG" AND PUBLISHER="BPB";
+--------------------------+--------------+
| TITLE                    | AUTHOR       |
+--------------------------+--------------+
| ACING YOUR CS BOARD EXAM | PRANSHU AJAY |
+--------------------------+--------------+
1 row in set (0.00 sec)
```

c) To display list of all books wit PRICE more than 130 in acending order of QUANTITY



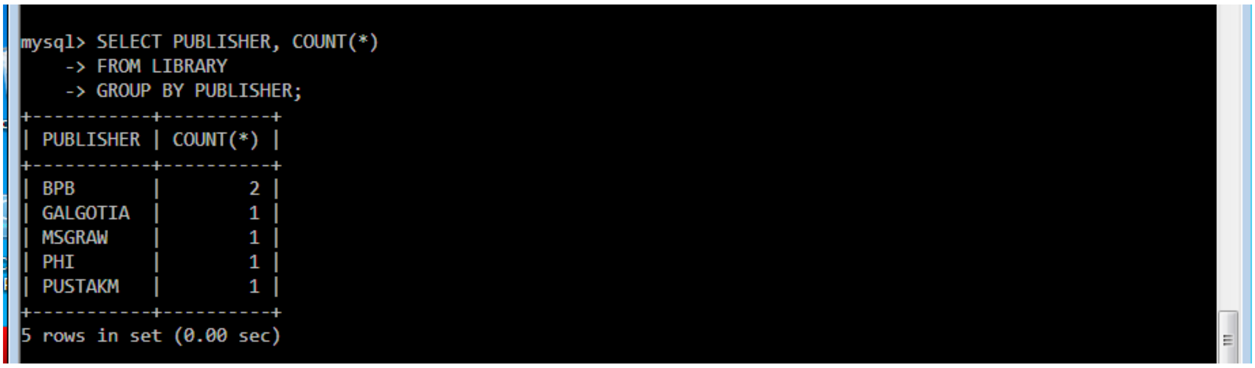
```
mysql> SELECT *
    -> FROM LIBRARY
    -> WHERE PRICE>130
    -> ORDER BY QUANTITY ASC;
+------+--------------------------+--------------+---------+-----------+----------+-------+
| NO   | TITLE                    | AUTHOR       | SUBJECT | PUBLISHER | QUANTITY | PRICE |
+------+--------------------------+--------------+---------+-----------+----------+-------+
|    5 | MASTERING WINDOWS        | COWART       | OS      | BPB       |        1 |   225 |
|    6 | ACING YOUR CS BOARD EXAM | PRANSHU AJAY | PROG    | BPB       |        1 |   425 |
|    2 | DOS GUIDE                | NORTON       | OS      | PHI       |        3 |   175 |
|    1 | DATA STRUCTURE           | LIPSCHUTE    | OS      | MSGRAW    |        4 |   217 |
|    3 | TURBO C++                | ROBERT LAFORE | PROG   | GALGOTIA  |        5 |   270 |
+------+--------------------------+--------------+---------+-----------+----------+-------+
5 rows in set (0.00 sec)
```

d) To display the average PRICE of PROG type books.



```
mysql> SELECT AVG(PRICE)
    -> FROM LIBRARY
    -> WHERE SUBJECT="PROG";
+------------+
| AVG(PRICE) |
+------------+
|   347.5000 |
+------------+
1 row in set (0.00 sec)
```

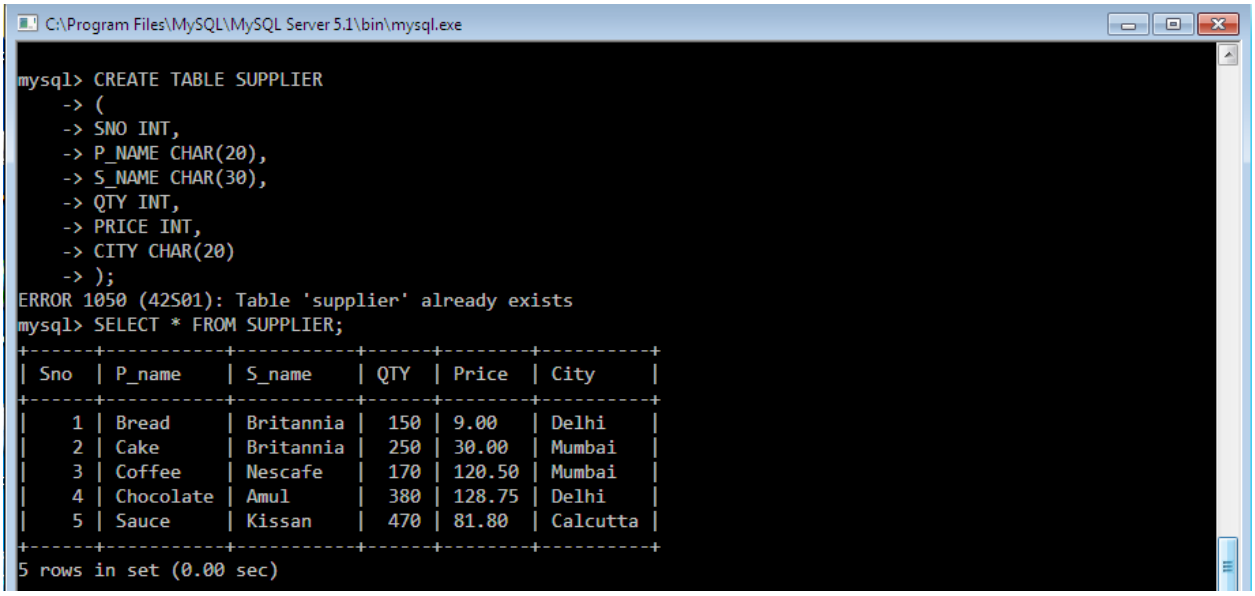
e) To display the PUBLISHER and the number of books of each PUBLISHER in table LIBRARY.



```
mysql> SELECT PUBLISHER, COUNT(*)
    -> FROM LIBRARY
    -> GROUP BY PUBLISHER;
+-----------+----------+
| PUBLISHER | COUNT(*) |
+-----------+----------+
| BPB       |        2 |
| GALGOTIA  |        1 |
| MSGRAW    |        1 |
| PHI       |        1 |
| PUSTAKM   |        1 |
+-----------+----------+
5 rows in set (0.00 sec)
```

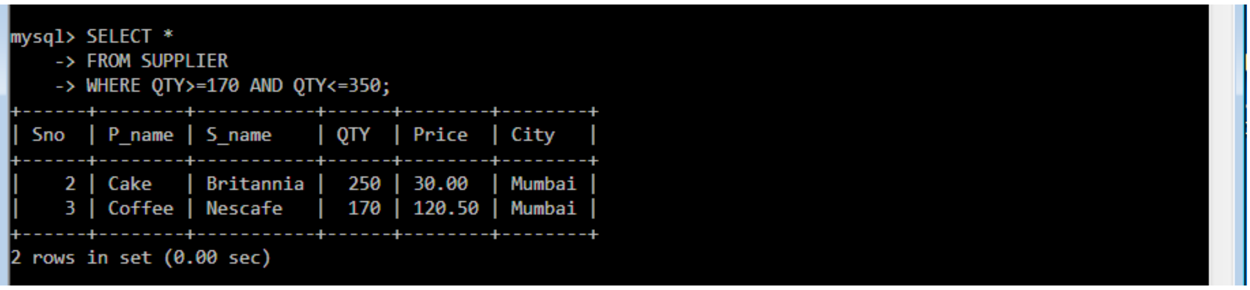
3)Create the following table in database "db".Write the SQL commands for the following queries:

Relation : SUPPLIER



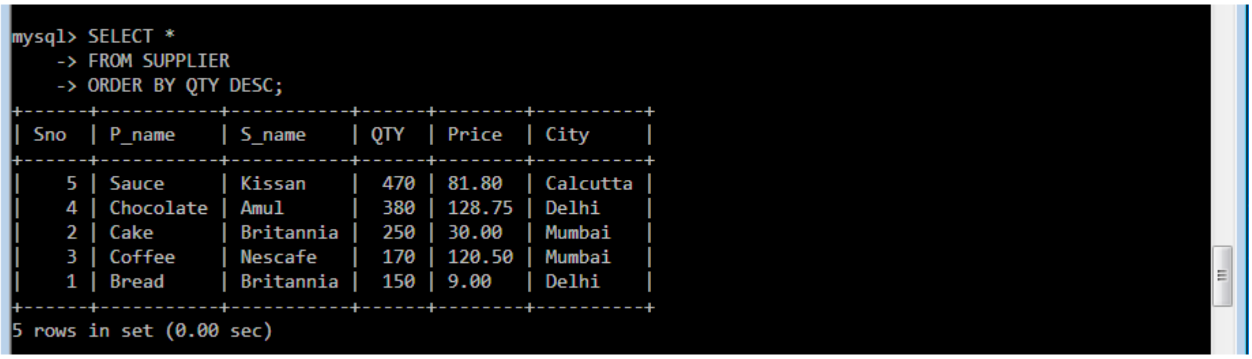
```
mysql> CREATE TABLE SUPPLIER
    -> (
    -> SNO INT,
    -> P_NAME CHAR(20),
    -> S_NAME CHAR(30),
    -> QTY INT,
    -> PRICE INT,
    -> CITY CHAR(20)
    -> );
ERROR 1050 (42S01): Table 'supplier' already exists
mysql> SELECT * FROM SUPPLIER;
+------+-----------+-----------+------+--------+----------+
| Sno  | P_name    | S_name    | QTY  | Price  | City     |
+------+-----------+-----------+------+--------+----------+
|    1 | Bread     | Britannia |  150 | 9.00   | Delhi    |
|    2 | Cake      | Britannia |  250 | 30.00  | Mumbai   |
|    3 | Coffee    | Nescafe   |  170 | 120.50 | Mumbai   |
|    4 | Chocolate | Amul      |  380 | 128.75 | Delhi    |
|    5 | Sauce     | Kissan    |  470 | 81.80  | Calcutta |
+------+-----------+-----------+------+--------+----------+
5 rows in set (0.00 sec)
```

a) Display data for all products whose QUANTITY is between 170 and 350.



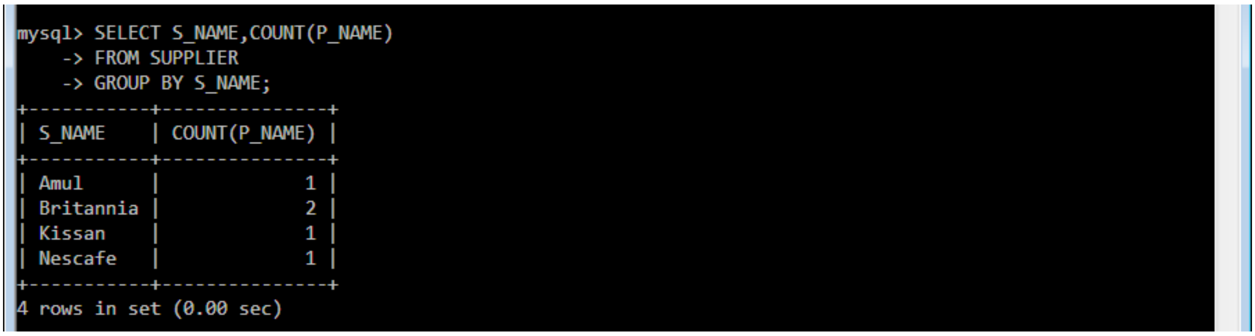
```
mysql> SELECT *
    -> FROM SUPPLIER
    -> WHERE QTY>=170 AND QTY<=350;
+------+--------+-----------+------+--------+--------+
| Sno  | P_name | S_name    | QTY  | Price  | City   |
+------+--------+-----------+------+--------+--------+
|    2 | Cake   | Britannia |  250 | 30.00  | Mumbai |
|    3 | Coffee | Nescafe   |  170 | 120.50 | Mumbai |
+------+--------+-----------+------+--------+--------+
2 rows in set (0.00 sec)
```

b) Display data for all products sorted by their QUANTITY in descending order.



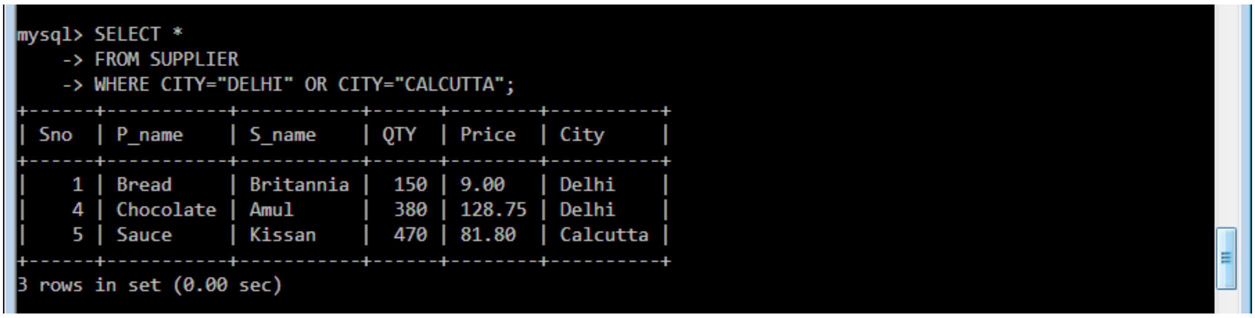
```
mysql> SELECT *
    -> FROM SUPPLIER
    -> ORDER BY QTY DESC;
+------+-----------+-----------+------+--------+----------+
| Sno  | P_name    | S_name    | QTY  | Price  | City     |
+------+-----------+-----------+------+--------+----------+
|    5 | Sauce     | Kissan    |  470 | 81.80  | Calcutta |
|    4 | Chocolate | Amul      |  380 | 128.75 | Delhi    |
|    2 | Cake      | Britannia |  250 | 30.00  | Mumbai   |
|    3 | Coffee    | Nescafe   |  170 | 120.50 | Mumbai   |
|    1 | Bread     | Britannia |  150 | 9.00   | Delhi    |
+------+-----------+-----------+------+--------+----------+
5 rows in set (0.00 sec)
```

c) To display Supplier name and the number of Products for each supplier.



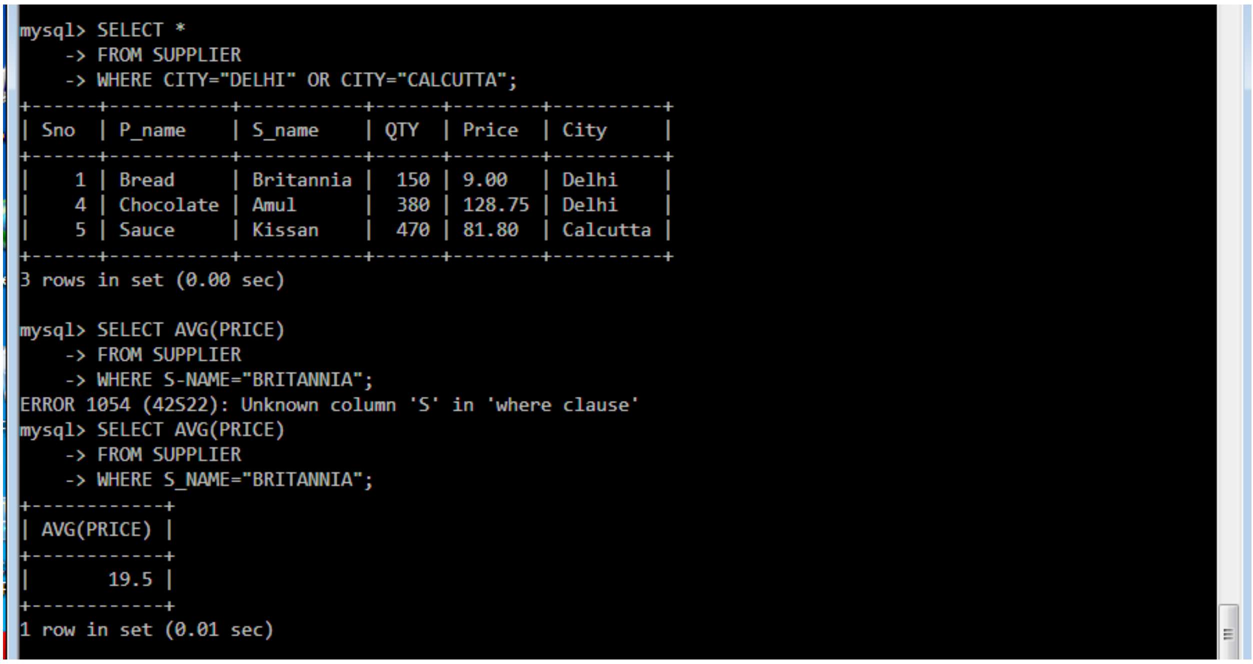
```
mysql> SELECT S_NAME,COUNT(P_NAME)
    -> FROM SUPPLIER
    -> GROUP BY S_NAME;
+-----------+---------------+
| S_NAME    | COUNT(P_NAME) |
+-----------+---------------+
| Amul      |             1 |
| Britannia |             2 |
| Kissan    |             1 |
| Nescafe   |             1 |
+-----------+---------------+
4 rows in set (0.00 sec)
```

d) To display all the products which are either in “DELHI” or in “CALCUTTA” city.



```
mysql> SELECT *
    -> FROM SUPPLIER
    -> WHERE CITY="DELHI" OR CITY="CALCUTTA";
+------+-----------+-----------+------+--------+----------+
| Sno  | P_name    | S_name    | QTY  | Price  | City     |
+------+-----------+-----------+------+--------+----------+
|    1 | Bread     | Britannia |  150 | 9.00   | Delhi    |
|    4 | Chocolate | Amul      |  380 | 128.75 | Delhi    |
|    5 | Sauce     | Kissan    |  470 | 81.80  | Calcutta |
+------+-----------+-----------+------+--------+----------+
3 rows in set (0.00 sec)
```

e) To display the average PRICE of the BRITANNIA products.



```
mysql> SELECT *
    -> FROM SUPPLIER
    -> WHERE CITY="DELHI" OR CITY="CALCUTTA";
+------+-----------+----------+------+--------+----------+
| Sno  | P_name    | S_name   | QTY  | Price  | City     |
+------+-----------+----------+------+--------+----------+
|    1 | Bread     | Britannia |  150 | 9.00   | Delhi    |
|    4 | Chocolate | Amul     |  380 | 128.75 | Delhi    |
|    5 | Sauce     | Kissan   |  470 | 81.80  | Calcutta |
+------+-----------+----------+------+--------+----------+
3 rows in set (0.00 sec)

mysql> SELECT AVG(PRICE)
    -> FROM SUPPLIER
    -> WHERE S-NAME="BRITANNIA";
ERROR 1054 (42S22): Unknown column 'S' in 'where clause'
mysql> SELECT AVG(PRICE)
    -> FROM SUPPLIER
    -> WHERE S_NAME="BRITANNIA";
+------------+
| AVG(PRICE) |
+------------+
|       19.5 |
+------------+
1 row in set (0.01 sec)
```

`