INDEX

C++ Program List:

Program No: 1

Declare and define a class EMPLOYEE having the following members. Also write a menu based program to implement this class for 5 employees.

private :

name, basic

da( ) // returns DA, which is 60% of the BASIC.

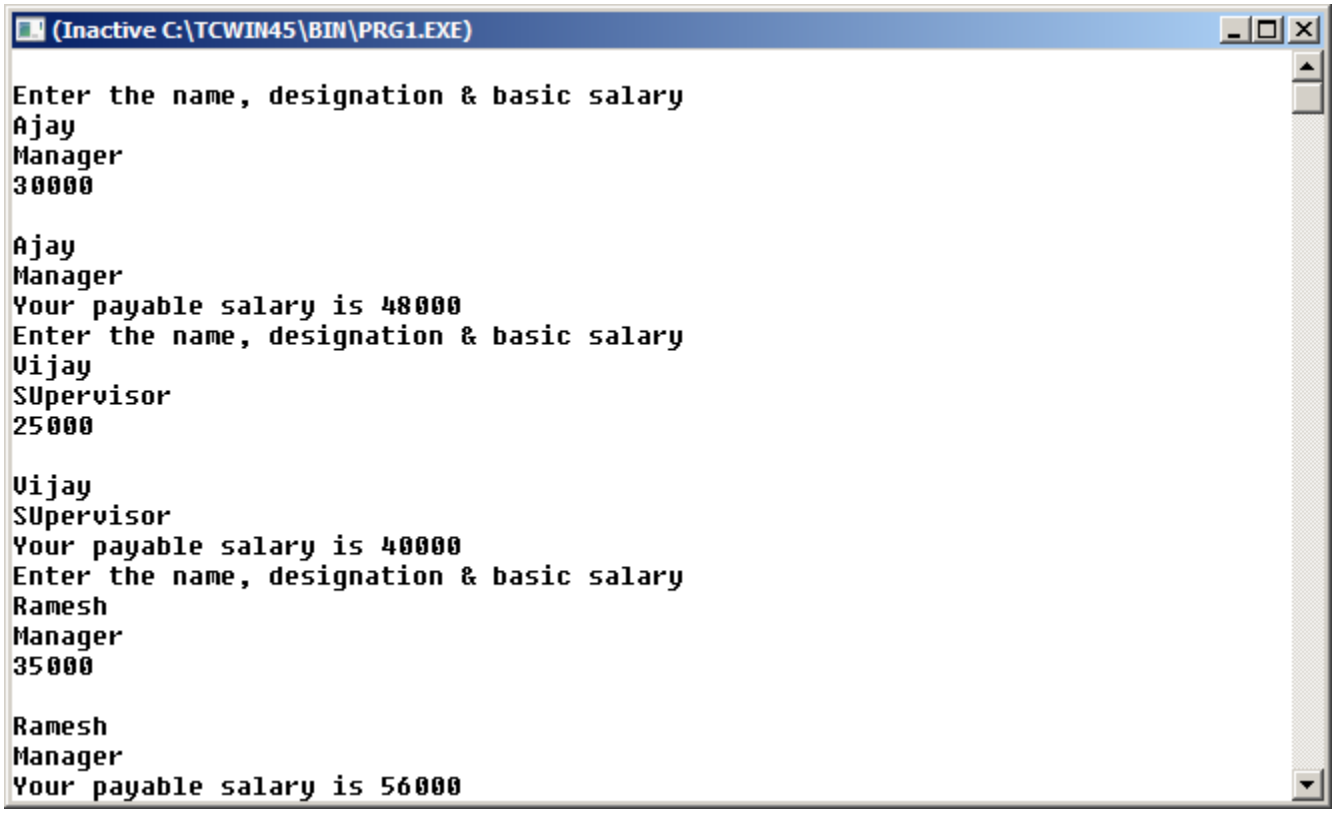
public :

getdata( )

showdata( )

```
//    Program No. 1

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

class employee
{
     char name[20];
     char desig[20];
     float basesal;
     float da()
     {
          float d=(basesal*3)/5;
          return d;
     }
     float cal_salary()
     {
          float d=da();
          float t;
          t=basesal+d;
          return t;
     }

 public:
     void input()
     {
          gets(name);
          gets(desig);
          cin>>basesal;
     }
```

```
      void output()
      {
            float t;
            cout<<endl<<name<<endl<<desig<<endl;
            t=cal_salary();
            cout<<"Your payable salary is "<<t;
      }
};


void main()
{
      employee E[5];
      for(int i=0;i<5;i++)
      {
            cout<<"\nEnter the name, designation & basic salary\n";
            E[i].input();
            E[i].output();
      }
}
```



Program 2:
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( )

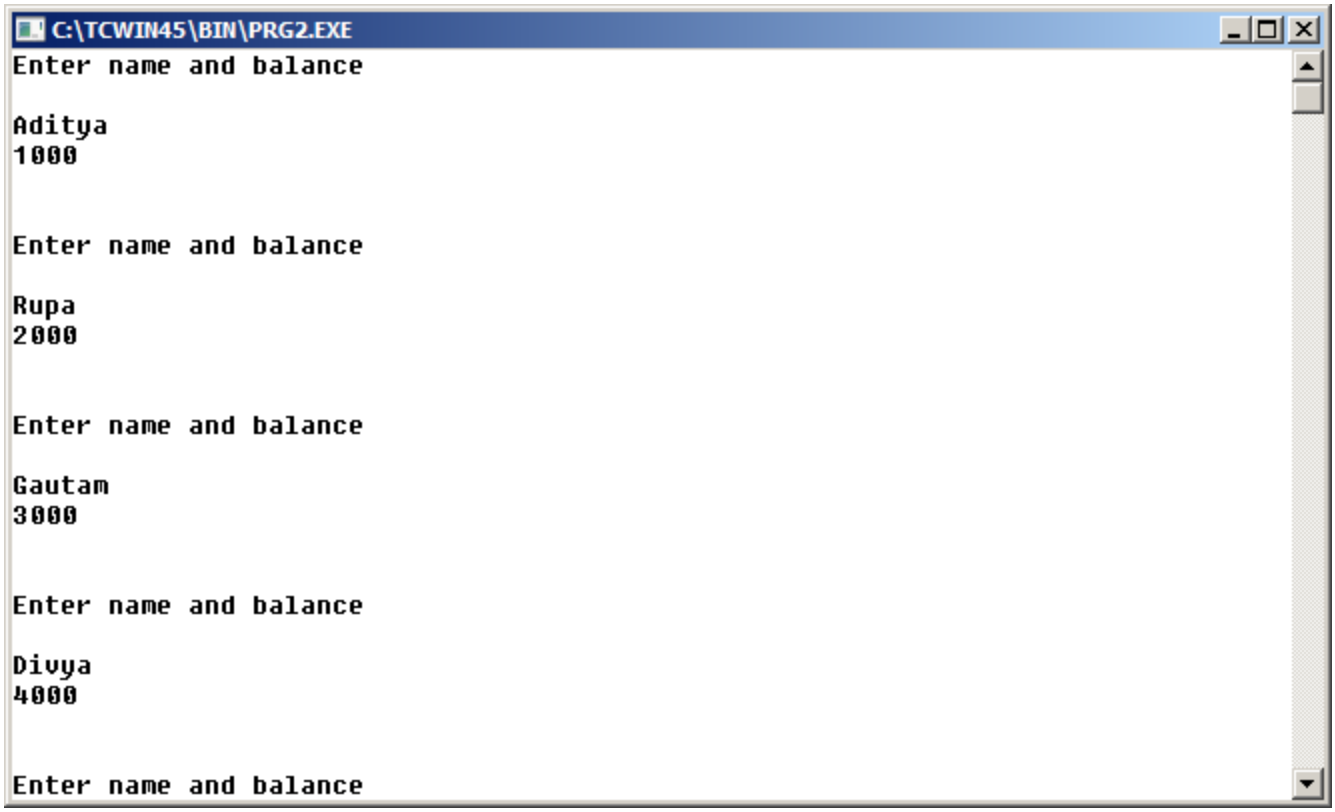
public:

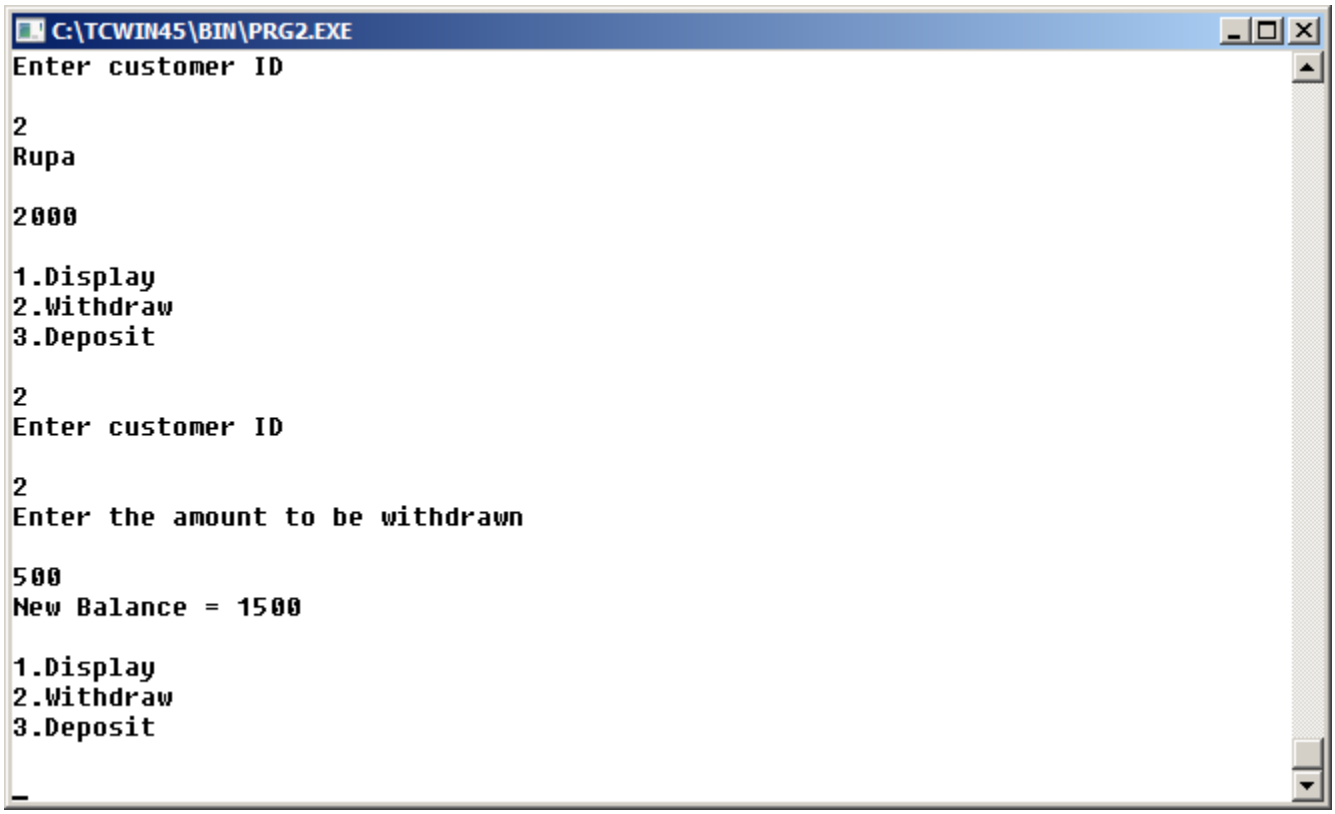
getdata( )

showdata( )

```
//   Program No. 2

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

class bank
{
     char name[20];
     float balance;
  public:
     void Getdata()
     {
          cout<<"Enter name and balance\n\n";
          gets(name);
          cin>>balance;
          cout<<endl<<endl;
     }
     void Showdata()
     {
          cout<<name<<endl<<endl;
          cout<<balance<<endl<<endl;
     }
     void Deposit()
     {
          float amt;
          cout<<"Enter the amount to be deposited\n\n";
          cin>>amt;
          balance=balance+amt;
          cout<<"New Balance = "<<balance<<endl<<endl;
     }
```

```
      void Withdraw()
      {
            float amt;
            cout<<"Enter the amount to be withdrawn\n\n";
            cin>>amt;
            if(amt>balance)
                  cout<<"Not enough balance\n\n";
             else
                  balance=balance-amt;
            cout<<"New Balance = "<<balance<<endl<<endl;
      }
};

void main()
{
            int i,cho;
            bank b[5];
            for(i=1;i<=5;i++)
                  b[i].Getdata();
            do
            {
                  cout<<"1.Display\n"<<"2.Withdraw\n"<<"3.Deposit\n\n";
                  cin>>cho;
                  switch(cho)
                  {
                  case 1:   cout<<"Enter customer ID\n\n";
                                    cin>>i;
                                    b[i].Showdata();
                                    break;
                  case 2:   cout<<"Enter customer ID\n\n";
                                    cin>>i;
                                    b[i].Withdraw();
                                    break;
                  case 3:   cout<<"Enter customer ID\n\n";
                                    cin>>i;
                                    b[i].Deposit();
                                    break;
                  default: cout<<"Wrong ID\n\n";
                  }
            } while(cho!=4);
}
```


C:\TCWIN45\BIN\PRG2.EXE
Enter name and balance
Aditya
1000
Enter name and balance
Rupa
2000
Enter name and balance
Gautam
3000
Enter name and balance
Divya
4000
Enter name and balance


C:\TCWIN45\BIN\PRG2.EXE
Enter customer ID
2
Rupa
2000
1.Display
2.Withdraw
3.Deposit
2
Enter customer ID
2
Enter the amount to be withdrawn
500
New Balance = 1500
1.Display
2.Withdraw
3.Deposit

Program 3:
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>=200else "FAIL"

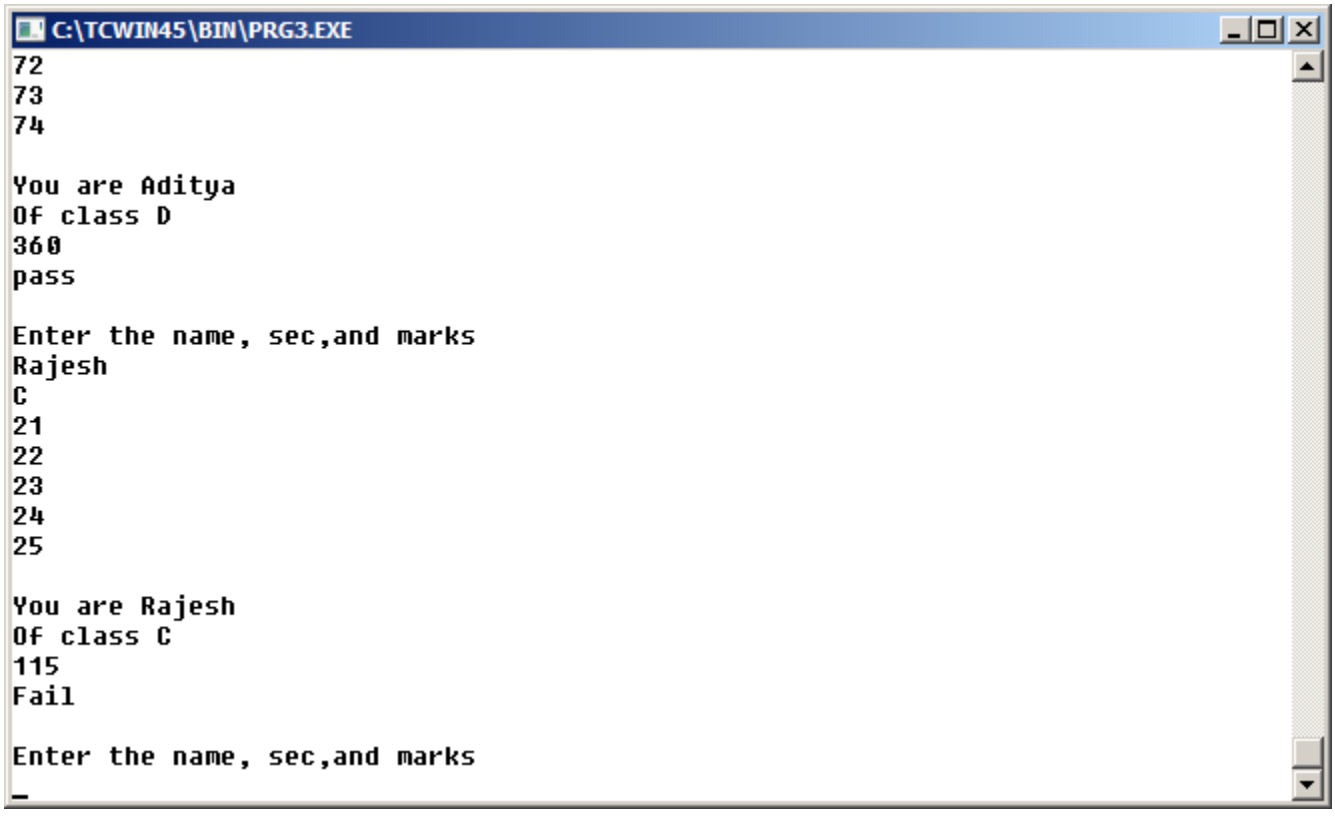
public:

getdata( )

showdata( )

```
//    Program No. 3



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

class student
{
     char name[20];
     char Class[5];
     int marks[5];
     void result()
     {
          float t=0;
          for(int i=0;i<5;i++)
               t=t+marks[i];
          cout<<t<<endl;
          if(t>=200)
               cout<<"pass";
          else
               cout<<"Fail";
     }
 public:
     void input()
     {
          gets(name);
          gets(Class);
          for(int j=0;j<5;j++)
                 cin>>marks[j];

     }
```

```
      void output()
      {
            int x;
            cout<<endl<<"You are "<<name<<endl;
            cout<<"Of class "<<Class<<endl;
            result();

      }
} ;


void main()
{
      student s[10];
      for(int i=0;i<10;i++)
      {
            cout<<"Enter the name, sec,and marks\n";
            s[i].input();
            s[i].output();
            cout<<endl<<endl;
      }
}
```

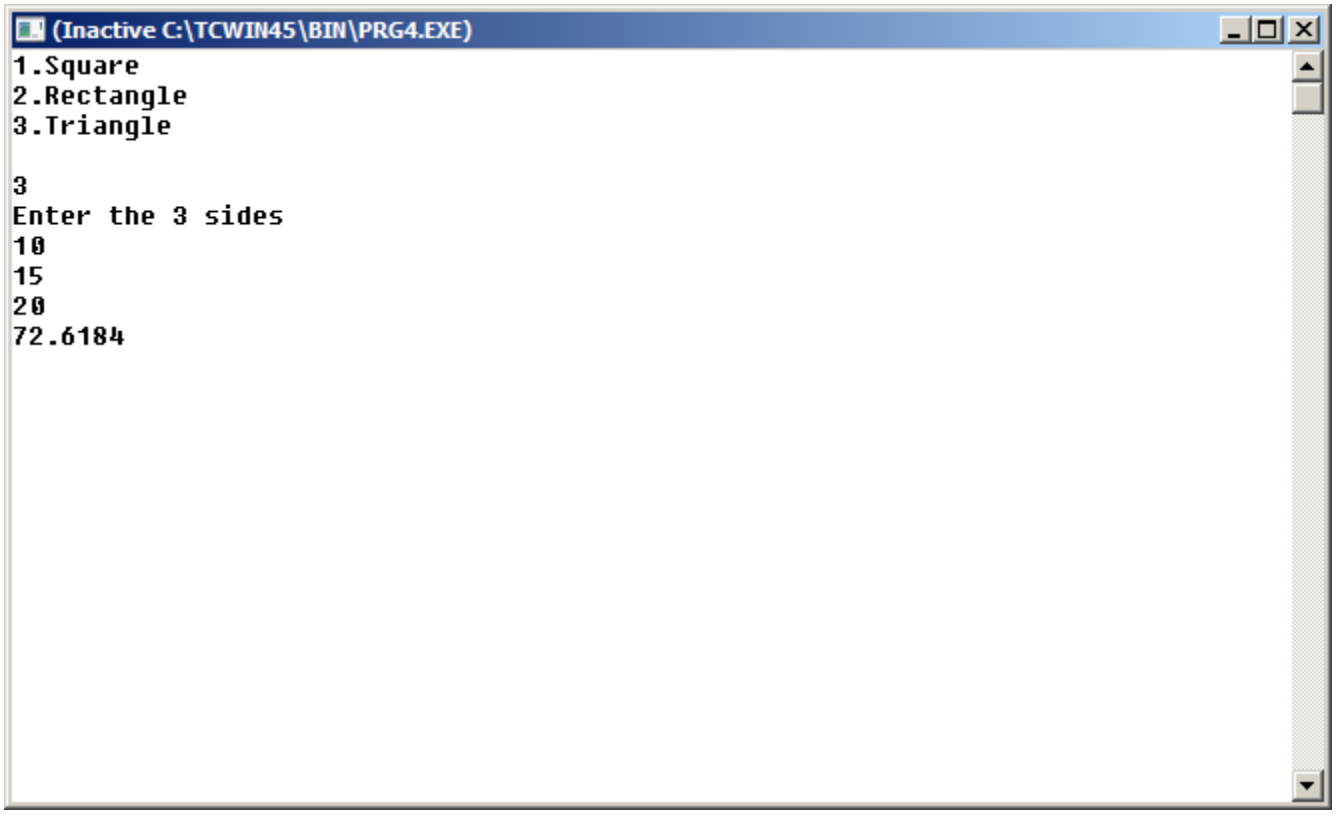


Program 4:

Declare and define a function having name area( ) and three definitions to calculate area of square, rectangle and triangle (using HERO's formulae). Write the complete menu based to use these functions effectively.

```
//    Program No. 4



#include<iostream.h>
#include<math.h>

void area(float s)                              //Area of a Square
{
     float areas=s*s;
     cout<<areas;
}

void area(float l,float b)                      //Area of a Rectangle
{
     float arear=l*b;
     cout<<arear;
}

void area(float p,float q,float r)              //Area of a Triangle
{
     float s=(p+q+r)/2;
     float areat=pow(s*(s-p)*(s-q)*(s-r),0.5);
     cout<<areat;
}

void main()
{
     float s,l,b,p,q,r;

     cout<<"1.Square\n2.Rectangle\n3.Triangle\n\n";
     int cho;
     cin>>cho;
     switch(cho)
     {
          case 1:   cout<<"Enter the side\n";
                    cin>>s;
                    area(s);
                    break;

          case 2:   cout<<"Enter the length and breadth\n";
                    cin>>l>>b;
                    area(l,b);
                    break;
```

```
            case 3:   cout<<"Enter the 3 sides\n";
                            cin>>p>>q>>r;
                            area(p,q,r);
                            break;
        }
}
```

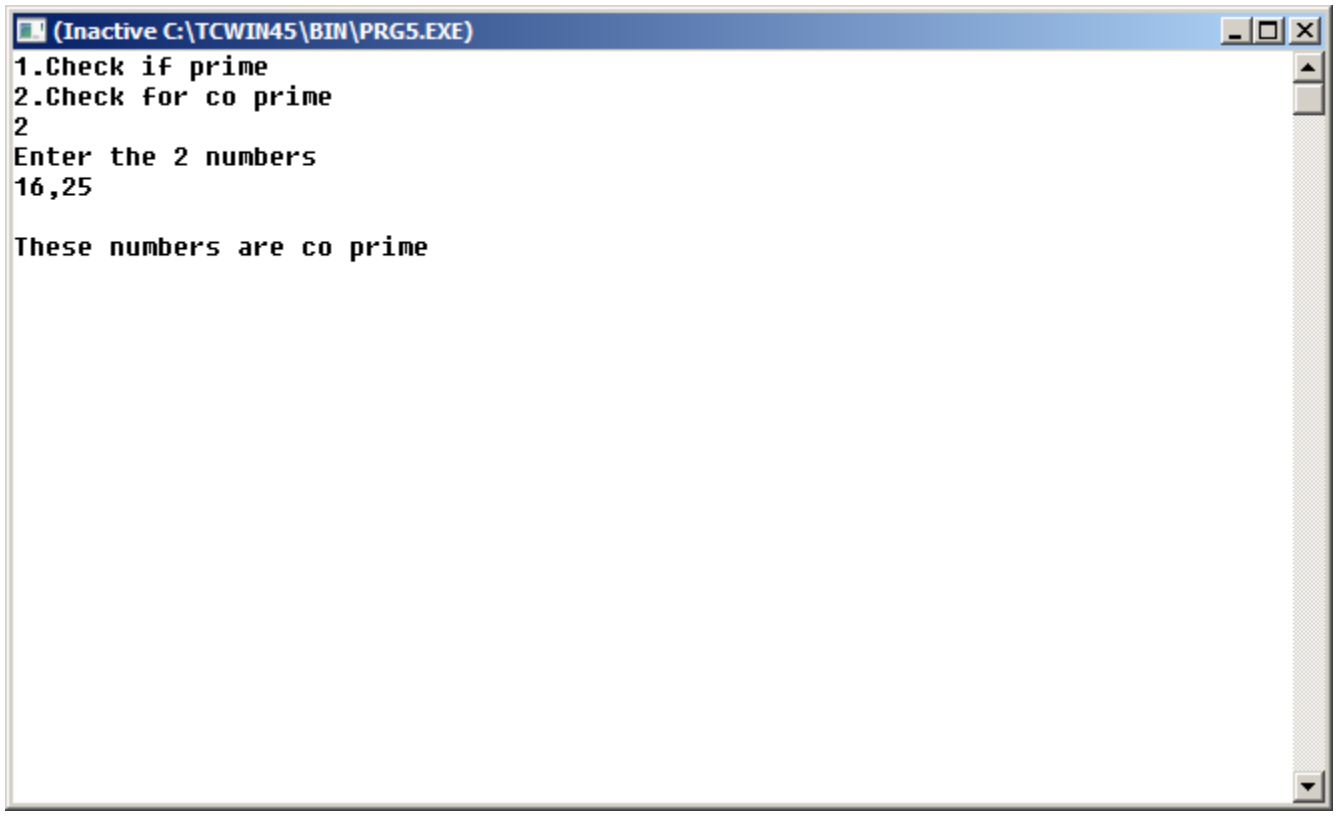


Program 5:
Define the following functions having same name PRIME( ) and also write the required program to demonstrate their use.

a. **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.**
b. **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.**

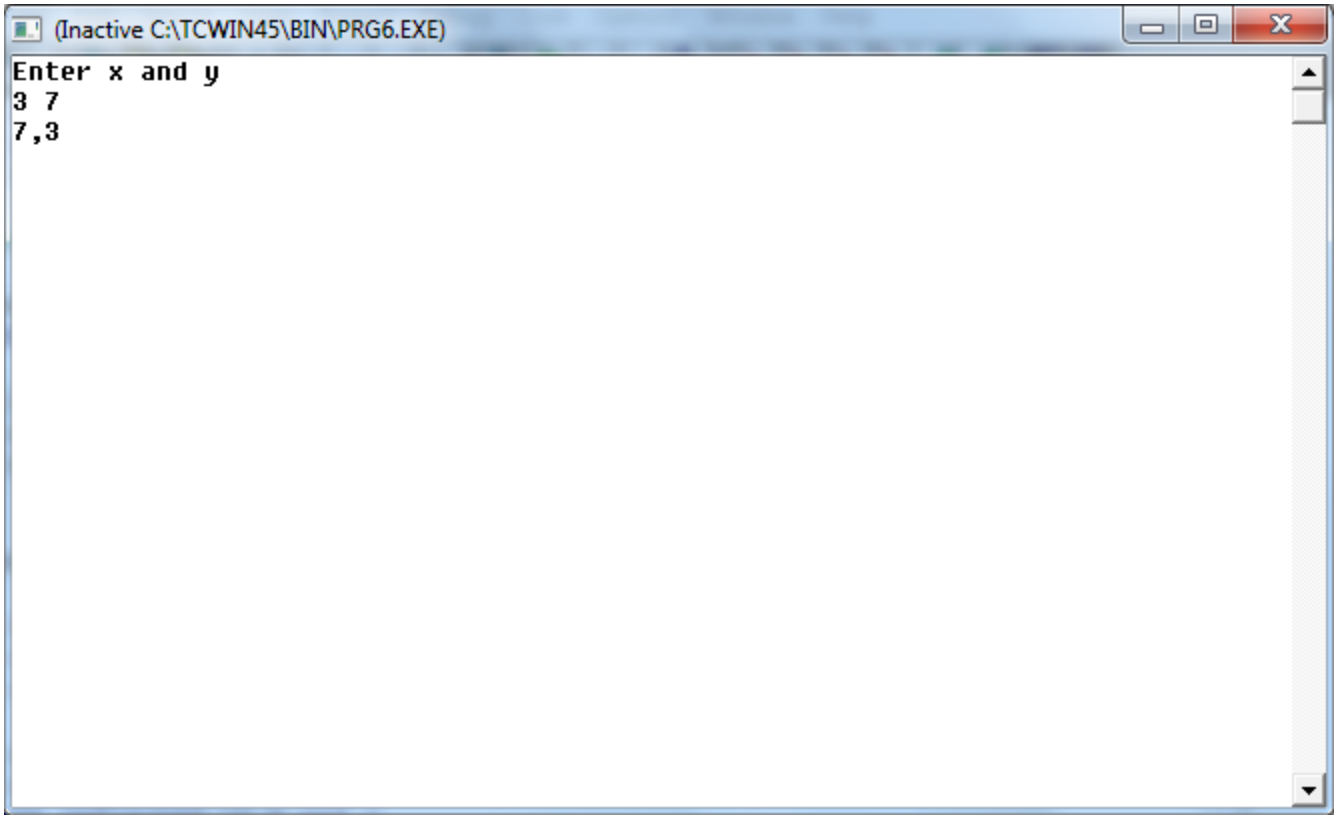
```
//    Program No. 5



#include<iostream.h>

void isprime(int p)
{
      int f,i;
      f=1;
      for(i=2;i<p;i++)
      {
            if(p%i==0)
                  f=0;
      }
      cout<<"\nThe given number is a prime\n";
}


void isprime(int a,int b)
{
      int hcf;
      for(int i=1;i<=a&&i<=b;i++)
      {
            if((a%i==0)&&(b%i==0))
                  {hcf=i;}
      }

  cout<<"\nThese numbers are co prime\n";
}

void main()
{
      int cho,p,a,b;
      cout<<"1.Check if prime\n2.Check for co prime\n";
      cin>>cho;

      switch(cho)
```

```
    {
        case 1:   cout<<"Enter the number\n";
                  cin>>p;
                  isprime(p);
                  break;

        case 2:   cout<<"Enter the 2 numbers\n";
                  cin>>a>>b;
                  isprime(a,b);
                  break;
    }
}
```



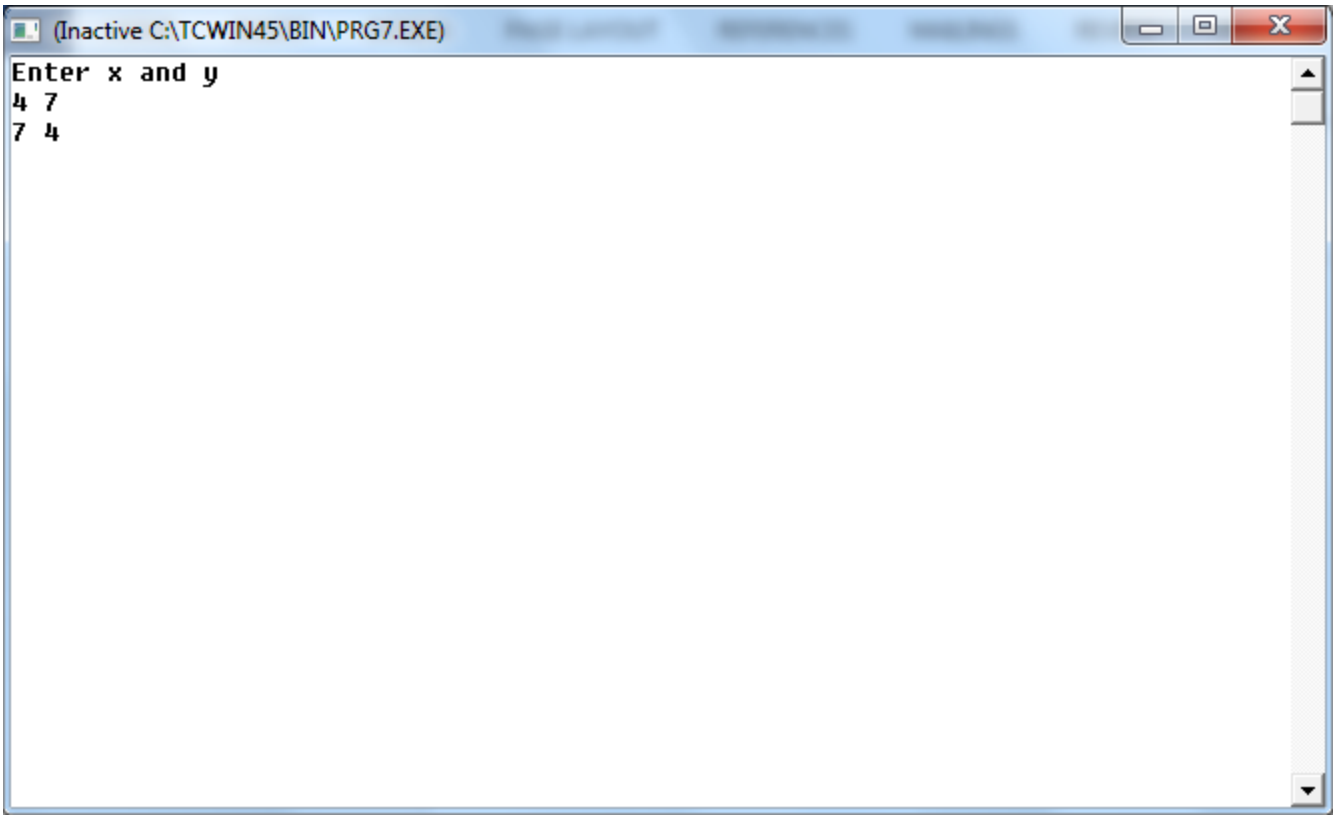
Program 6:
SWAP using CALL BY REFERENCE.

```
//    Program No. 6



#include<iostream.h>

void swap(int &a,int &b)        //No new variables created
                                //OR No fresh memory is allocated
{
      int c=a;
      a=b;
      b=c;
}

void main()
{
      int x,y;
      cout<<"Enter x and y\n";
      cin>>x>>y;

      swap(x,y) ;
      cout<<x<<','<<y<<endl;    //Changes in formal arguments a and b
                                //are reflected in x and y
}
```



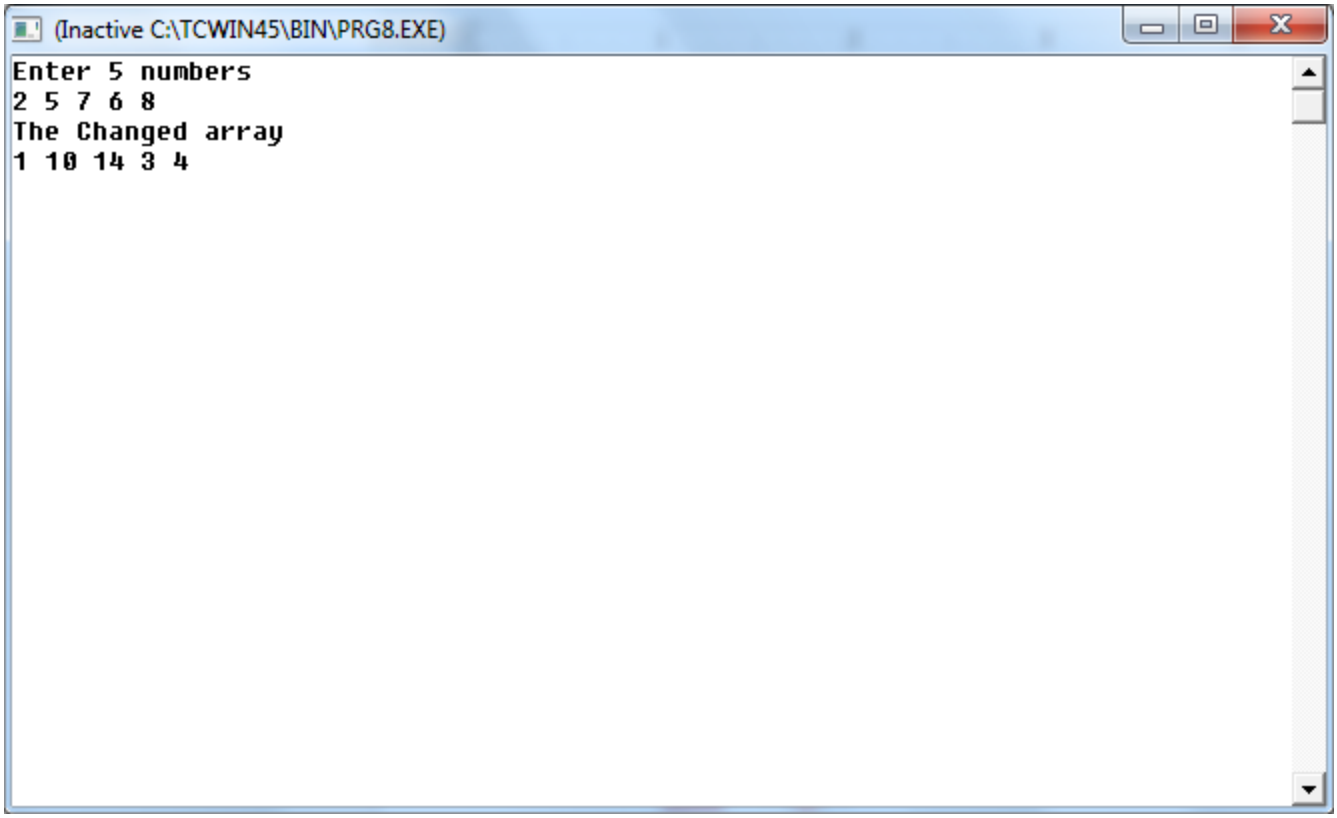
Program 7:
SWAP using CALL BY POINTERS.

```
//    Program No. 7



#include<iostream.h>

void swap(int *a,int *b) //Pointers as the formal arguments
{
      int c=*a;
      *a=*b;
      *b=c;
}

void main()
{
      int x,y;
      cout<<"Enter x and y\n";
      cin>>x>>y;

      swap(&x,&y) ;          //Call by pointers
      cout<<x<<' '<<y<<endl;//Changes are reflected in x and y

}
```



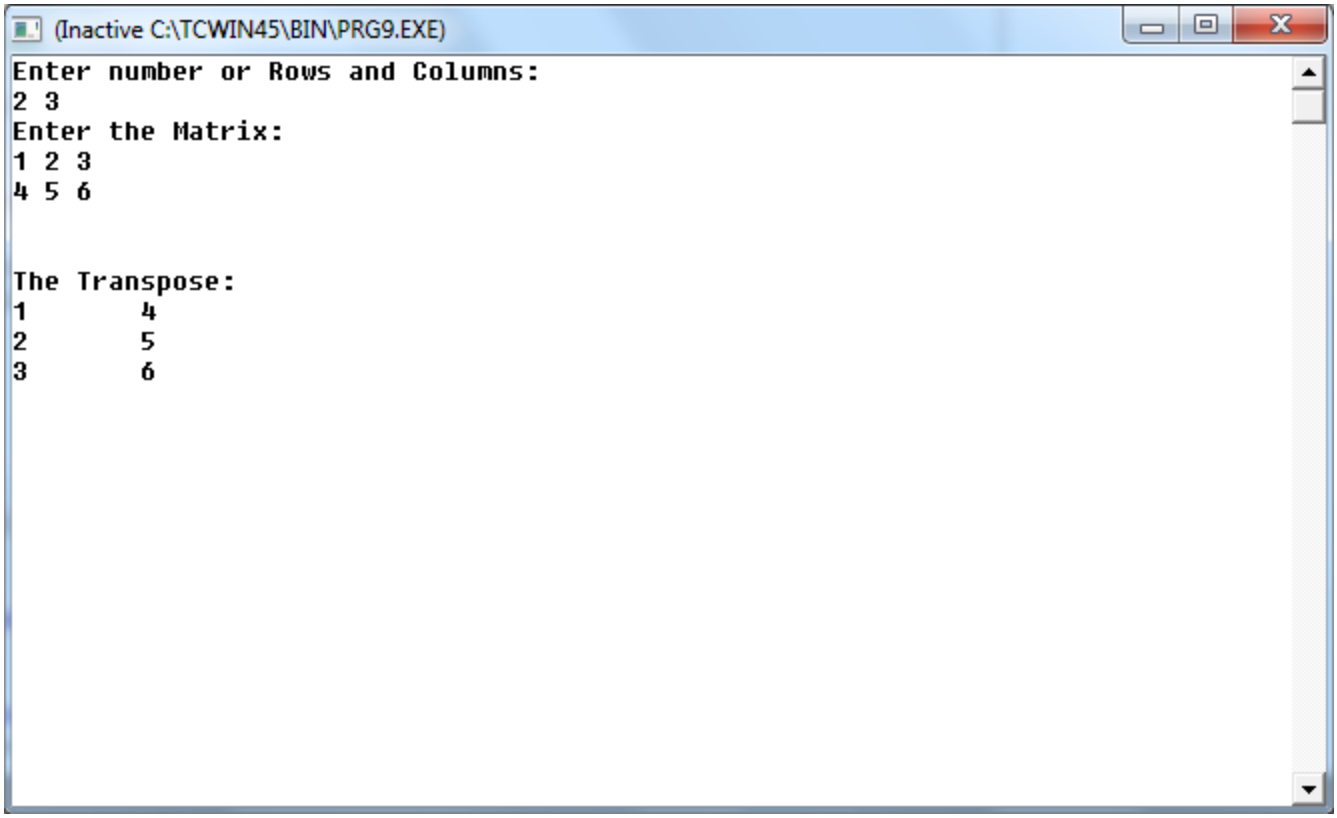
Program 8:
1 D Array - Replace an even number with half value and an odd with double value.

```
//    Program No. 8



#include<iostream.h>
const N=5;
void Change(int a[],int N)
{
      for(int i=0;i<N;i++)
      {
            if(a[i]%2==0)
                  a[i]=a[i]/2;
             else
                  a[i]=2*a[i];
      }
}

void main()
{
      int i, x[5];
      cout<<"Enter 5 numbers\n";
      for(i=0;i<N;i++)
            cin>>x[i];

      Change(x,5);
      cout<<"The Changed array"<<endl;
      for(i=0;i<N;i++)
            cout<<x[i]<<' ';
      cout<<endl;
}
```


(Inactive C:\TCWIN45\BIN\PRG8.EXE)
Enter 5 numbers
2 5 7 6 8
The Changed array
1 10 14 3 4

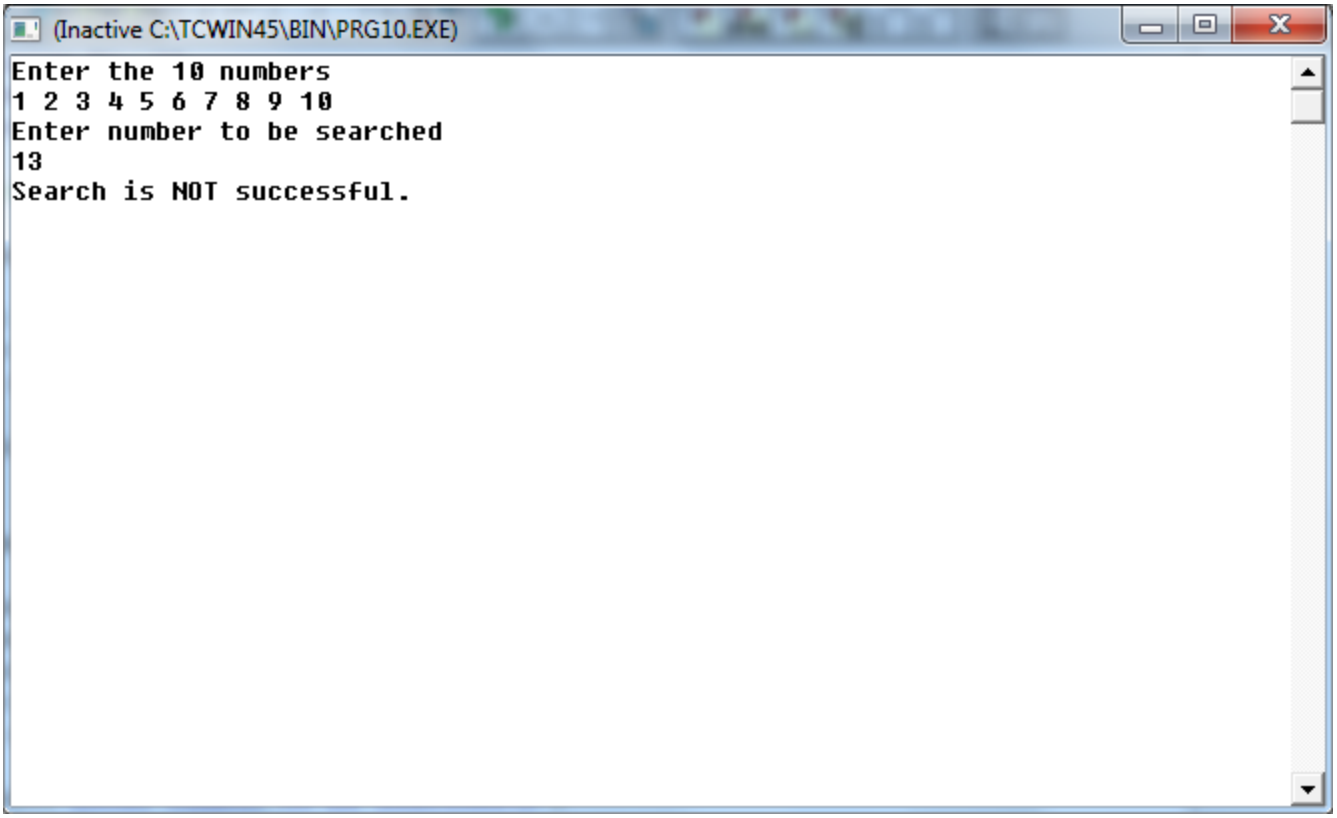
Program 9:
2 D Array – Transpose of a Matrix

```
//      Program No. 9



#include<iostream.h>

void Transpose(int M[][10],int R, int C)
{
        int i,j;
        for(j=0;j<C;j++)
        {
                for(i=0;i<R;i++)
                        cout<<M[i][j]<<'\t';
                cout<<endl;
         }
}

void main()
{
        int MAT[10][10];
        int i,j,R,C;

        cout<<"Enter number or Rows and Columns:\n";
        cin>>R>>C;

        cout<<"Enter the Matrix:\n";
        for(i=0;i<R;i++)
                for(j=0;j<C;j++)
                        cin>>MAT[i][j];

        cout<<"\n\nThe Transpose: \n";
        Transpose(MAT,R,C);
}
```


(Inactive C:\TCWIN45\BIN\PRG9.EXE)
Enter number or Rows and Columns:
2 3
Enter the Matrix:
1 2 3
4 5 6
The Transpose:
1 4
2 5
3 6

Program 10:
Define a function to search whether a float element DATA is present in a sorted array A[N] or not (use Sequential Search Technique). Also write a minimum program to invoke the function.

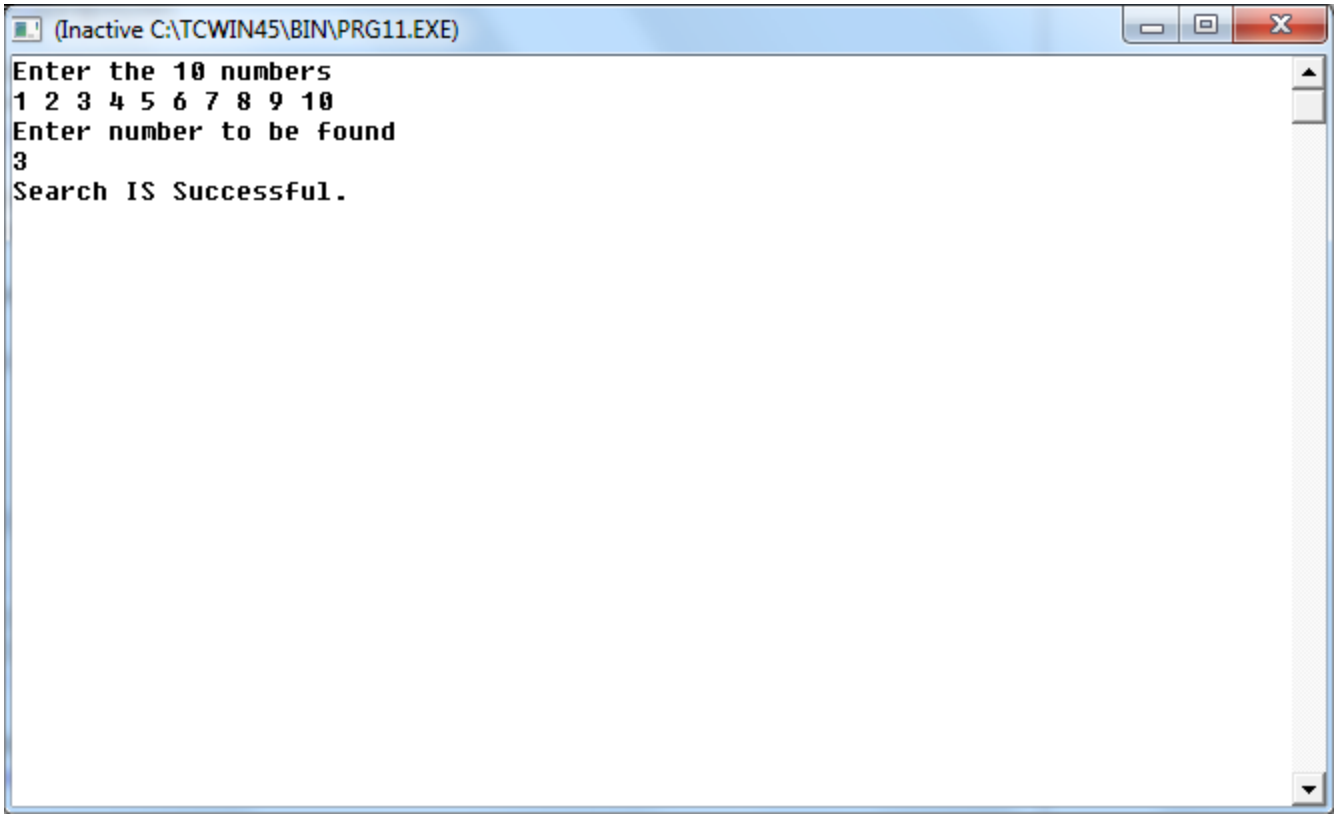
```
//    Program No. 10



#include<iostream.h>

int seqsearch(float A[],int N,float DATA)
{
      int Found =0;
      for(int i=0;i<N;i++)
            if(A[i]==DATA)
                  Found=1;
      return Found;
}

void main()
{
      float ARR[10] ;
      float DAT;
      cout<<"Enter the 10 numbers \n";
      for(int j=0;j<10;j++)
            cin>>ARR[j];

      cout<<"Enter number to be searched\n";
      cin>>DAT;

      if(seqsearch(ARR,10,DAT))
            cout<<"Search IS Successful.\n";
       else
            cout<<"Search is NOT successful.\n";

}
```


(Inactive C:\TCWIN45\BIN\PRG10.EXE)
Enter the 10 numbers
1 2 3 4 5 6 7 8 9 10
Enter number to be searched
13
Search is NOT successful.

Program 11:

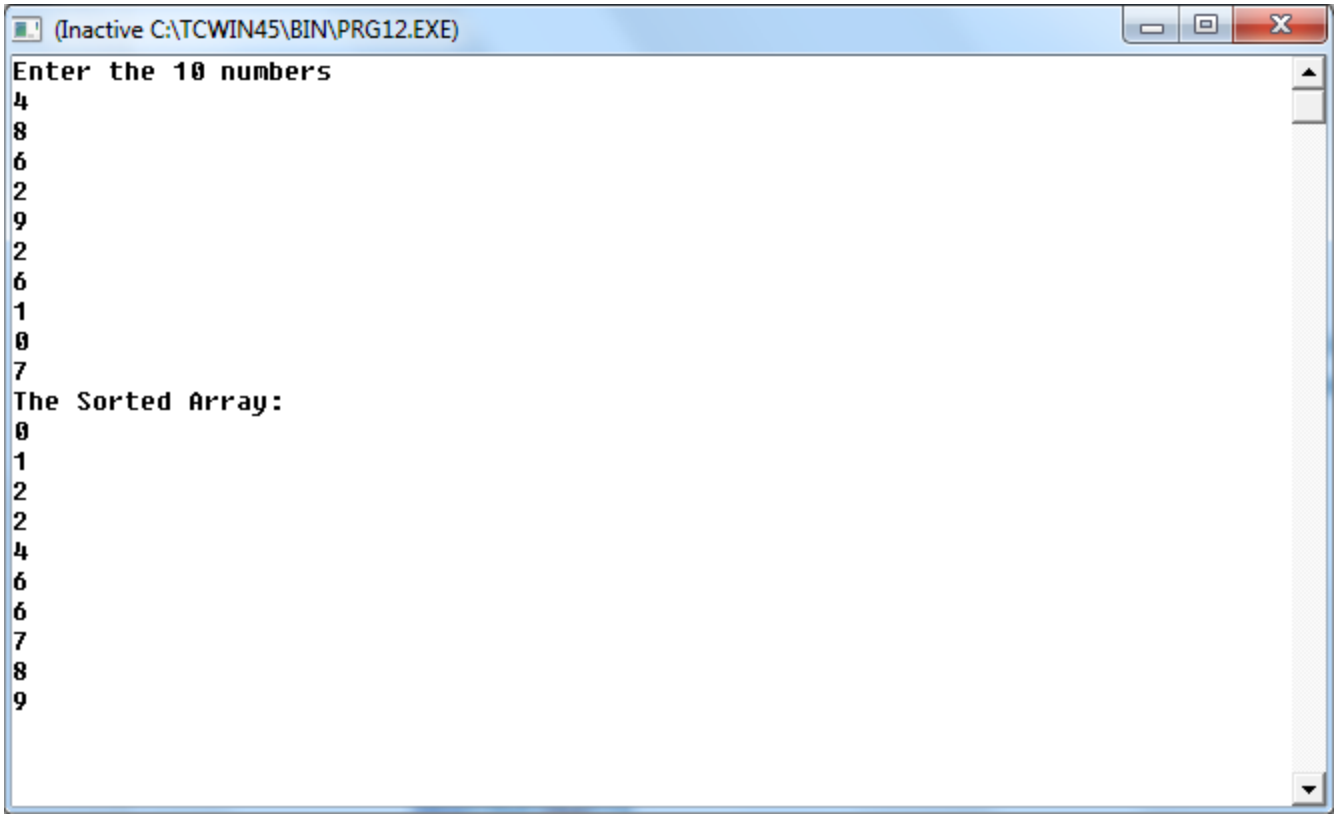
Define a function to search whether a float element DATA is present in a sorted array A[N] or not (use Binary Search Technique). Also write a minimum program to invoke the function.

```
//    Program No. 11



#include<iostream.h>

int binasearch(float A[],int N, float DATA)
{
      int Low=0,Up=N-1,Mid,Found=0;
      while(Found==0&&Up>Low)
      {
            Mid=(Low+Up)/2;
            if(A[Mid]==DATA)
                  Found=1;
             else
                  if(A[Mid]>DATA)
                      Up=Mid-1;
                  else
                      Low=Mid+1;
      }
      return Found;
}

void main()
{
      float ARR[10] ;
      float DAT;
      cout<<"Enter the 10 numbers \n";
      for(int j=0;j<10;j++)
            cin>>ARR[j];
      cout<<"Enter number to be found\n";
      cin>>DAT;
      if(binasearch(ARR,10,DAT))
            cout<<"Search IS Successful.\n";
        else
            cout<<"Search is NOT successful.\n";
}
```


(Inactive C:\TCWIN45\BIN\PRG11.EXE)
Enter the 10 numbers
1 2 3 4 5 6 7 8 9 10
Enter number to be found
3
Search IS Successful.

Program 12:
Define a function to accept an array A[N] of float numbers as argument 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.

```
//    Program No. 12




#include<iostream.h>

void bubsort(float A[],int N)
{
      float Temp;
      for(int i=0;i<N-1;i++)
            for(int j=0;j<N-i-1;j++)
                  if(A[j]>A[j+1])
                  {
                        Temp=A[j];
                        A[j]=A[j+1];
                        A[j+1]=Temp;
                  }
}

void main()
{
      float ARR[10];
      cout<<"Enter the 10 numbers\n";

      for(int i=0;i<10;i++)
            cin>>ARR[i];

      bubsort(ARR,10);

   cout<<"The Sorted Array:\n";

      for(int k=0;k<10;k++)
            cout<<ARR[k]<<endl;
}
```

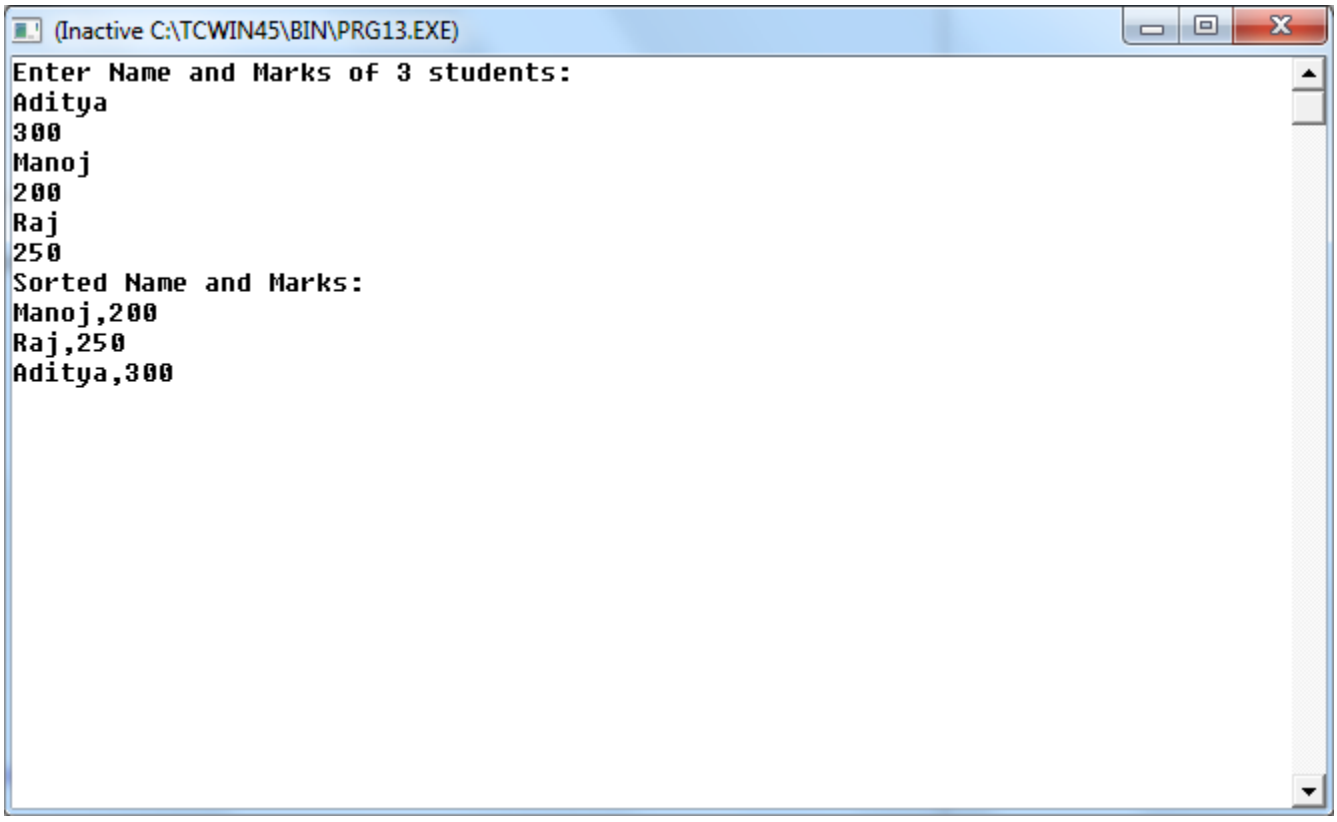

(Inactive C:\TCWIN45\BIN\PRG12.EXE)
Enter the 10 numbers
4
8
6
2
9
2
6
1
0
7
The Sorted Array:
0
1
2
2
4
6
6
7
8
9

Program 13:
Define a function to accept an array A[N] of type STUDENT as declared below and then return the same array but in ascending order on their Marks. (Use Selection Sort Technique). Also write a minimum C++ program to illustrate the defined function.

```
struct STUDENT
{
        char Name[20];
        int Marks;
};
```

```
//    Program No. 13



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

struct STUDENT
{
      char Name[20];
      int Marks;
};

void sel_sort(STUDENT A[], int N)
{
      int i,j,M;
      STUDENT Temp;
      for(i=0;i<N-1;i++)
      {
            M=i;
            for(j=i+1;j<N;j++)
            {
                  if(A[M].Marks>A[j].Marks)
                        M=j;
            }
            if(M!=i)
            {
                  Temp=A[M];
                  A[M]=A[i];
                  A[i]=Temp;
            }
      }
}
```

```
void main()
{
      STUDENT S[3];
      int i;
      cout<<"Enter Name and Marks of 3 students:\n";
      for(i=0;i<3;i++)
      {
            gets(S[i].Name);
            cin>>S[i].Marks;
      }
      sel_sort(S,3);
      cout<<"Sorted Name and Marks:\n";
      for(i=0;i<3;i++)
      {
            cout<<S[i].Name<<','<<S[i].Marks<<endl;
      }
}
```

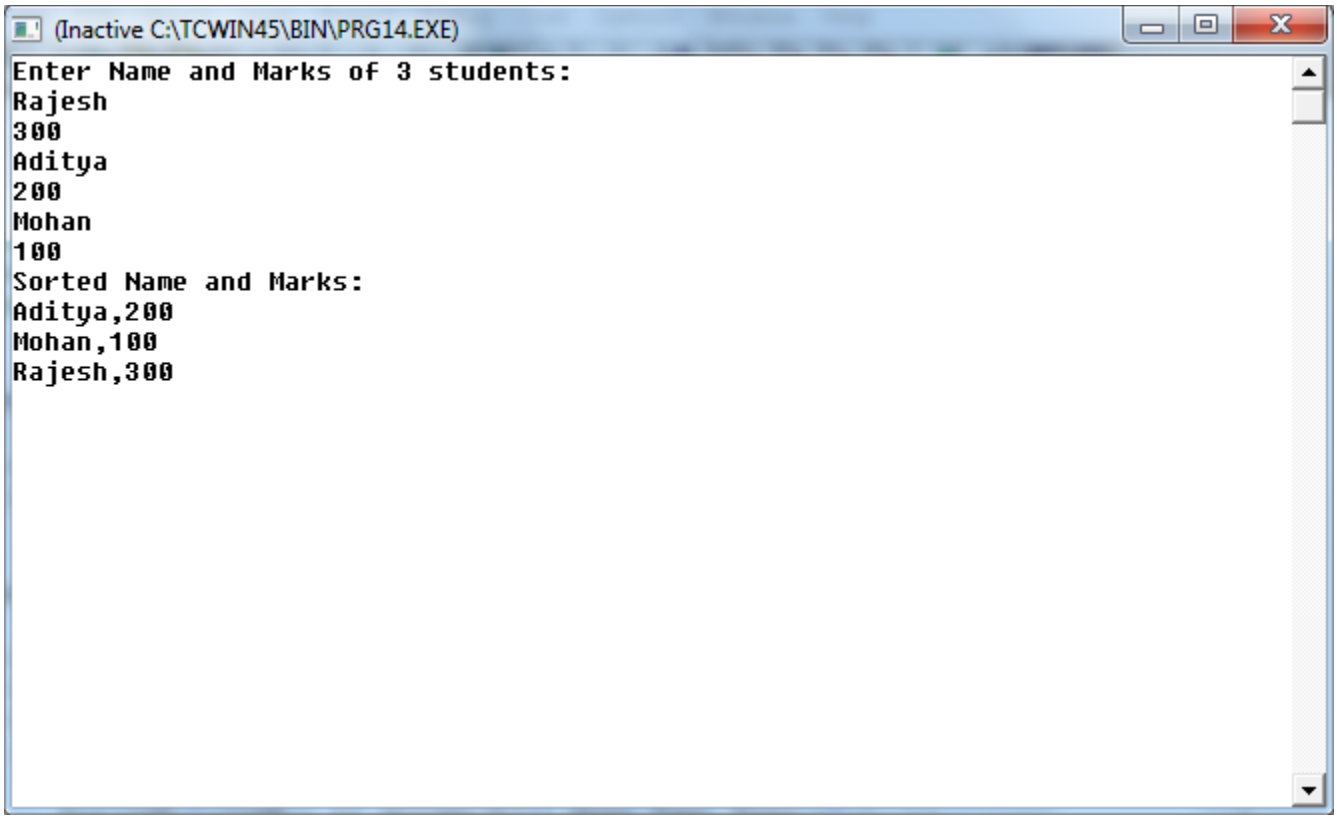


Program 14:
Define a function to accept an array A[N] of type STUDENT declared below and then return the same array but in ascending order. (Use Insertion Sort Technique). Also write a minimum C++ program to illustrate the defined function.

```
struct STUDENT
{
      char Name[20];
      int Marks;
};
```

```
//    Program No. 14



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


struct STUDENT
{
     char Name[20];
     int Marks;
};

void insert_sort(STUDENT A[], int N)
{
     int i,j,k;
     STUDENT Temp;
     for(i=0;i<N;i++)
     {
          Temp=A[0];
          for(j=N-1;j>=N-i && strcmp(A[j].Name,Temp.Name)>0;j--);
          for(k=0;k<j;k++)
               A[k]=A[k+1];
          A[j]=Temp;
     }
}

void main()
{
     STUDENT S[3];
     int i;
     cout<<"Enter Name and Marks of 3 students:\n";
     for(i=0;i<3;i++)
     {
          gets(S[i].Name);
          cin>>S[i].Marks;
     }
```

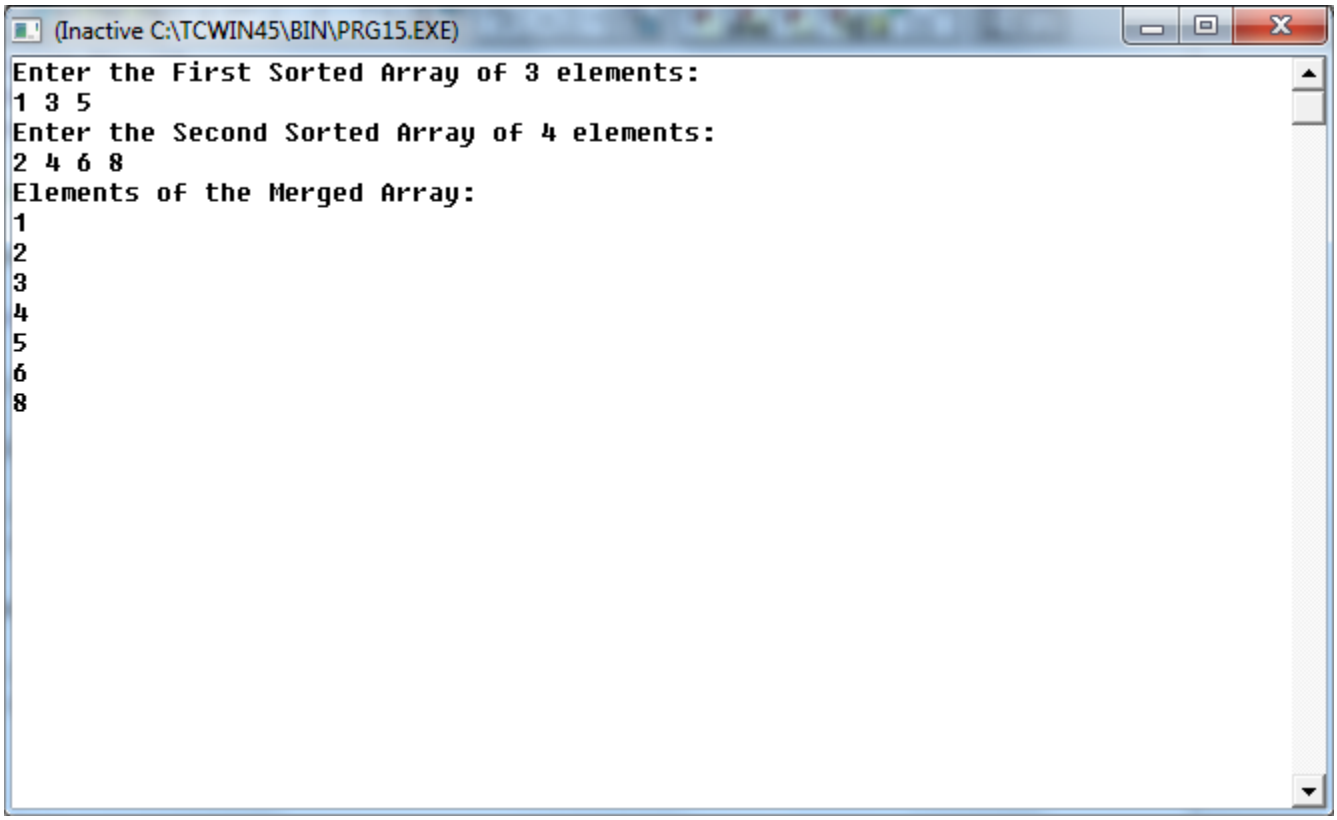
```
      insert_sort(S,3);
      cout<<"Sorted Name and Marks:\n";
      for(i=0;i<3;i++)
      {
            cout<<S[i].Name<<','<<S[i].Marks<<endl;
      }
}
```



Program 15:
WAF to accept two arrays A[M] and B[N] and then return the merged array C[]. (All the arrays are in ascending order).

```
//    Program No. 15



#include<iostream.h>

void merge(float A[],int M,float B[],int N,float C[])
{
      int i=0,j=0,k=0;

      while(i<M&&j<N)
      {
            if(A[i]<B[j])
            {
                  C[k]=A[i];
                  k++;i++;
            }
            else
            {
                  C[k]=B[j];
                  k++;j++;
            }
      }

      while(i<M)
      {
            C[k]=A[i];
            k++;i++;
      }

      while(j<N)
      {
            C[k]=B[j];
            k++;j++;
      }

}
```

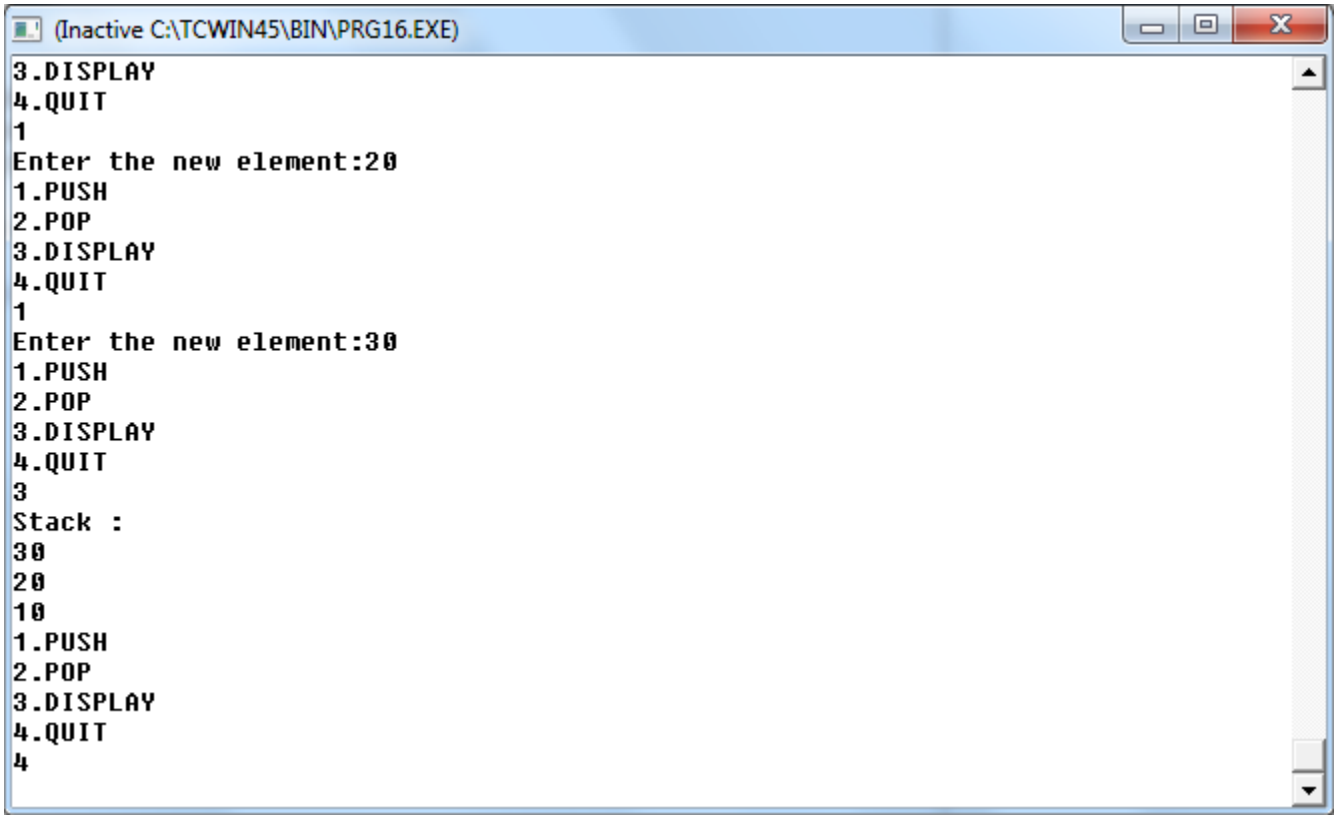
```
void main()
{
      float X[3],Y[4],Z[7];
      int i;
      cout<<"Enter the First Sorted Array of 3 elements:\n";
      for(i=0;i<3;i++)
            cin>>X[i];

      cout<<"Enter the Second Sorted Array of 4 elements:\n";
      for(i=0;i<4;i++)
            cin>>Y[i];

      cout<<"Elements of the Merged Array:\n";
      merge(X,3,Y,4,Z);
      for(i=0;i<7;i++)
            cout<<Z[i]<<endl;
}
```



Program 16:
Menu based ARRAY implementation of STACK.

```
//    Program No. 16



#include<iostream.H>
#include<conio.H>
const N=5;
class STACK
{
     float A[N];
     int top;
  public:
     STACK(){top=-1;}
     void PUSH();
     void POP();
     void DISPLAY();
};
void STACK::PUSH()
{
     if(top==N-1)                          //Checks overflow
          cout<<"Overflow,addition not possible\n";
      else
          top++;
      cout<<"Enter the new element:";
      cin>>A[top];
}
void STACK::POP()
{
     if(top==-1)                           //Checks underflow
          cout<<"Underflow! stack is empty\n";
      else
      {
          top--;
          cout<<"An element gets deleted.\n";
      }
}
void STACK::DISPLAY()
{
     if (top==-1)                          //Checks underflow
          cout<<"Stack is empty\n";
      else
      {
          cout<<"Stack :\n";
          for(int i=top;i>=0;i--)
               cout<<A[i]<<endl;
      }
}
```

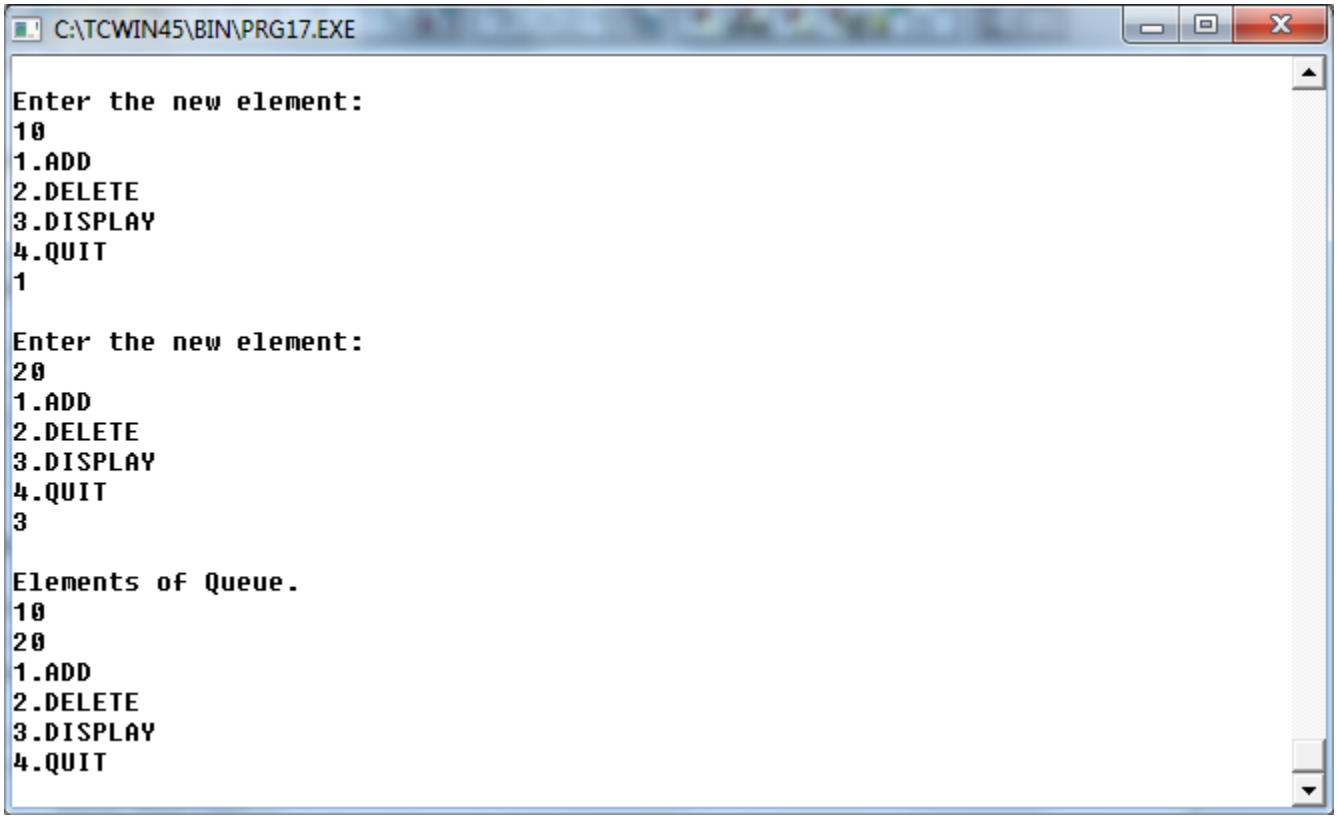
```
void main()
{
      int choice;
      STACK S;
      do
      {
            cout<<"1.PUSH\n";
            cout<<"2.POP\n";
            cout<<"3.DISPLAY \n";
            cout<<"4.QUIT\n";

            choice=getche();
       cout<<endl;
            switch(choice)
            {
                  case '1': S.PUSH();    break;
                  case '2': S.POP();    break;
                  case '3': S.DISPLAY();    break;
                  case '4':     break;
                  default :cout<<"\nWrong Choice Entered !!\n\n";
            }
      }while(choice!='4');
}
```



Program 17:
Menu based OOP for ARRAY implementation of QUEUE.

```
//    Program No. 17



#include<iostream.H>
#include<conio.H>
const N=5;
class queue
{
     float A[N];
     int rear ;
  public:
     queue(){rear=-1;}
     void addq();
     void delq();
     void dispq();
};
void queue::addq()             //Addition at the rear
{
     if(rear==N-1)
          cout<<"\nOverflow! Addition is NOT possible.\n";
      else
      {
          rear++;
      cout<<"\nEnter the new element:\n";
          cin>>A[rear];
      }
}
void queue::delq()             //Deletion at the front
{
     if(rear==-1)
          cout<<"\nUnderflow! Queue is empty.\n";
      else
      {
          rear--;
          for (int i=0;i<=rear;i++)
               A[i]=A[i+1];
          cout<<"\nAn element is deleted from the Queue.\n";
      }
}
void queue::dispq()            //Displays the Queue
{
     if (rear==-1)
          cout<<"\nQueue is empty.\n";
      else
      {
          cout<<"\nElements of Queue.\n";
```

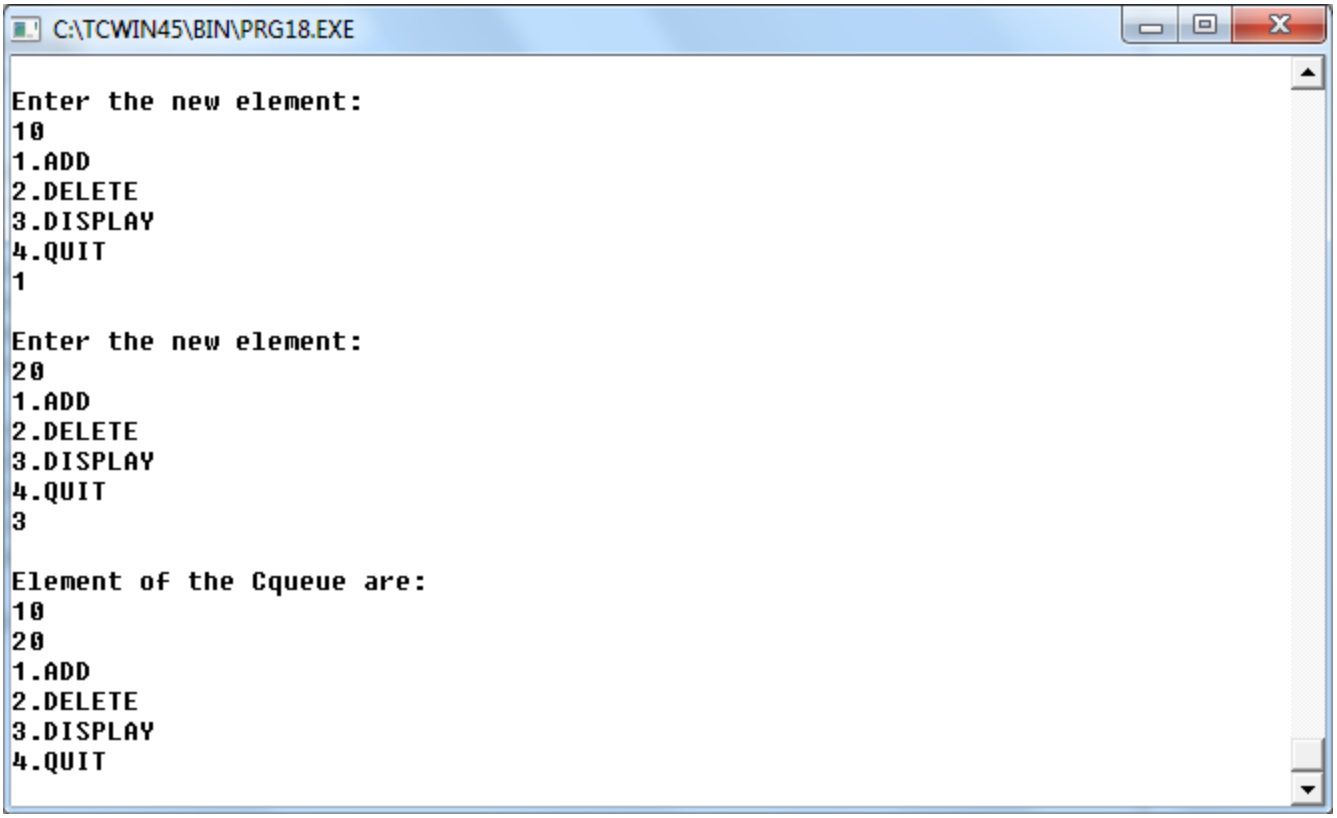
```
            for(int i=0;i<=rear;i++)
                  cout<<A[i]<<endl;
       }
}
void main()
{
      int choice;
      queue q;
      do
      {
            cout<<"1.ADD\n";
            cout<<"2.DELETE\n";
            cout<<"3.DISPLAY \n";
            cout<<"4.QUIT\n";
            choice=getche();
            cout<<endl;
            switch(choice)
            {
                  case '1': q.addq();    break;
                  case '2': q.delq();    break;
                  case '3': q.dispq();    break;
                  case '4':     break;
                  default :cout<<"\nWrong Choice Entered !!\n\n";
            }
      }while(choice!='4');
}
```



Program 18:
Menu based ARRAY implementation of CIRCULAR QUEUE.

```
//    Program No. 18



#include<iostream.H>
#include<conio.H>
const N=5;
class CQueue
{
      float A[N];
      int Front;
      int Rear ;
      public:
      CQueue(){Front=-1;Rear=-1;}
      void AddCQ();
      void DelCQ();
      void DispCQ();
};
void CQueue::AddCQ()                                  //Addition at the rear
{
      if((Front==0&&Rear==N-1)||(Front==Rear+1))    //Checks for Overflow
            cout<<"Overflow,addition not possible";
      else
      {
            Rear++;
            if (Rear==N)Rear=0;
            cout<<"\nEnter the new element:\n";
            cin>>A[Rear];
            if(Front==-1)Front=0;
      }
}
void CQueue::DelCQ()                              //Deletion at the front
{
      if(Front==-1||Rear==-1)
            cout<<"Underflow! CQueue is empty";
      else
      {
            cout<<"\nAn element deleted.\n";
            if(Front==Rear)                       //Only 1 element present
                  Front=Rear=-1;
            else
            {
                  Front++;
                  if(Front==N)Front=0;
            }
}}
```

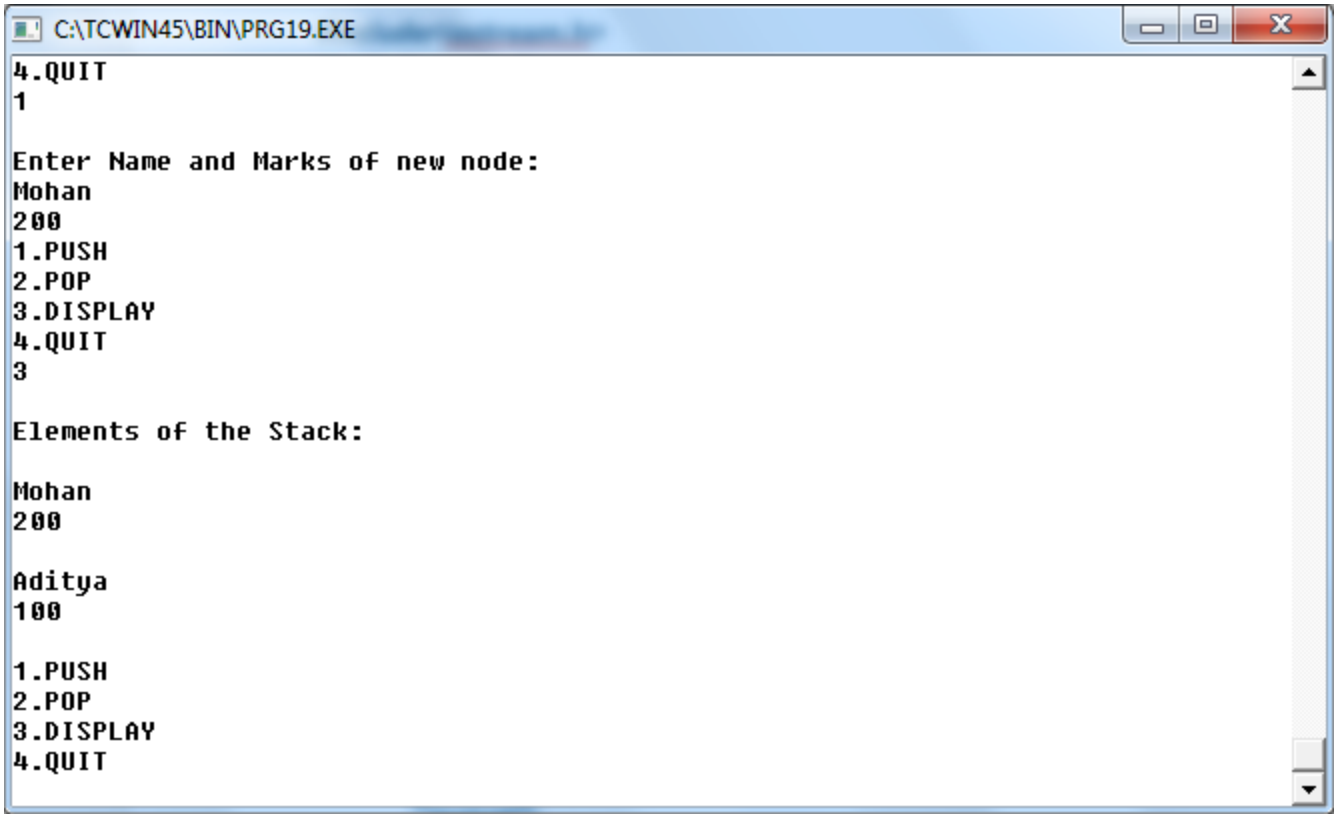
```
void CQueue::DispCQ()                          //Displays the Circular Queue
{
     if (Rear==-1||Front==-1)
          cout<<"CQueue is empty";
     else
     {
          cout<<"\nElement of the Cqueue are:\n";
          if(Front<=Rear)
          {
               for(int i=Front;i<=Rear;i++)
                    cout<<A[i]<<endl;
          }
          else
          {
               for(int k=Front;k<N;k++)
                    cout<<A[k]<<endl;

               for(int j=0;j<=Rear;j++)
                    cout<<A[j]<<endl;
          }
     }
}

void main()
{
     int choice;
     CQueue CQ;
     do
     {
          cout<<"1.ADD\n";
          cout<<"2.DELETE\n";
          cout<<"3.DISPLAY \n";
          cout<<"4.QUIT\n";
          choice=getche();
          cout<<endl;
          switch(choice)
          {
               case '1': CQ.AddCQ();    break;
               case '2': CQ.DelCQ();    break;
               case '3': CQ.DispCQ();    break;
               case '4':     break;
               default :cout<<"\nWrong Choice Entered !!\n\n";
          }
     }while(choice!='4');
}
```

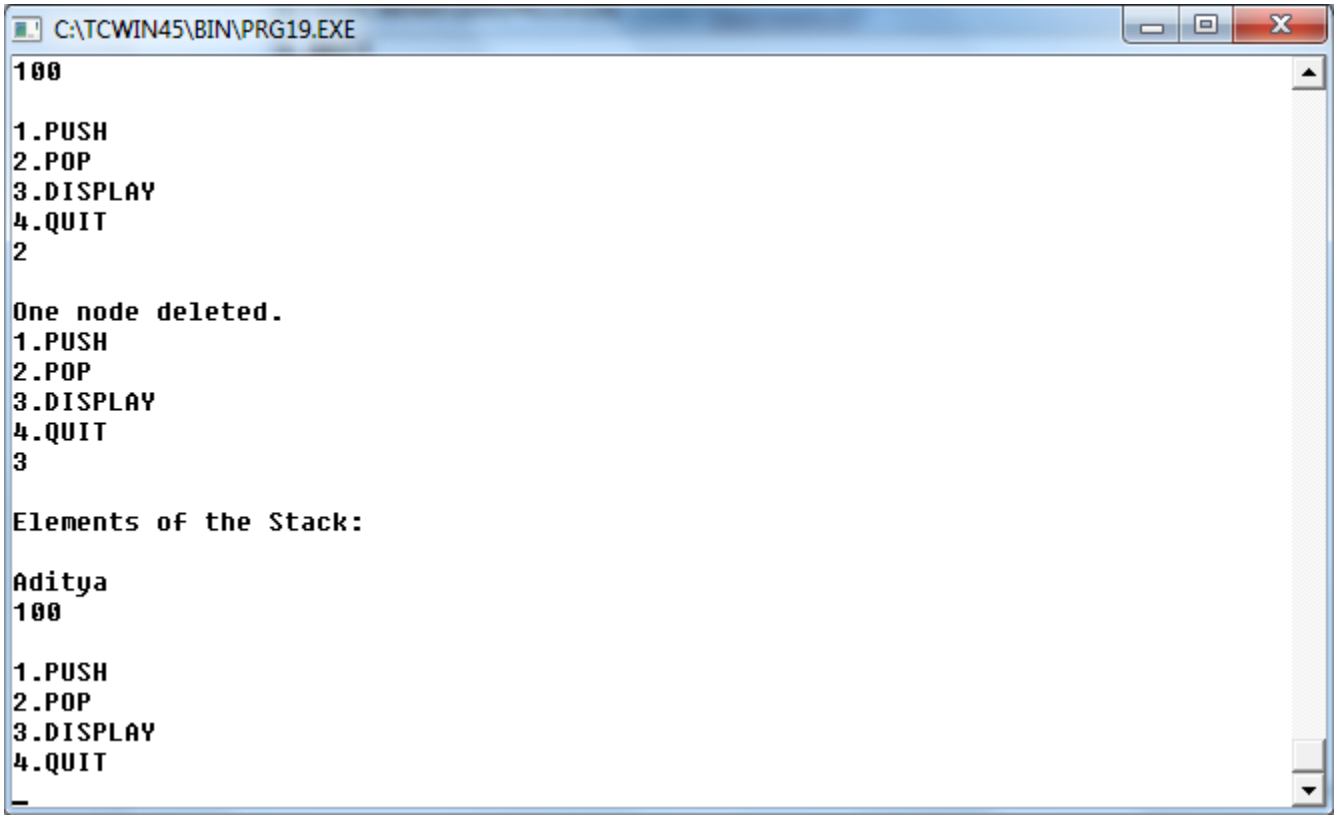

C:\TCWIN45\BIN\PRG18.EXE
Enter the new element:
10
1.ADD
2.DELETE
3.DISPLAY
4.QUIT
1
Enter the new element:
20
1.ADD
2.DELETE
3.DISPLAY
4.QUIT
3
Element of the Cqueue are:
10
20
1.ADD
2.DELETE
3.DISPLAY
4.QUIT

Program 19:
Linked list implementation of STACK, where each node consist of name & marks of students.

```
//    Program No. 19



#include<iostream.h>
#include<stdio.h>
#include<conio.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();
};

void Stack::Push()              //Addition of a node at the top
{
      Node *Temp;
      Temp=new Node;
      cout<<"\nEnter Name and Marks of new node:\n";
      gets(Temp->Name);
      cin>>Temp->Marks;
      Temp->Next=Top;
      Top=Temp;
}

void Stack::Pop()               //Deletion of the node at the top
{
      if(Top==NULL)
      cout<<"Underflow!!Stack is empty\n";
      else
      {
            Node *Temp;
            Temp=Top;
            Top=Top->Next;
            delete(Temp);
```

```
            cout<<"\nOne node deleted.\n";
      }
}

void Stack::Display()               //Displays the Stack
{
      if(Top==NULL)
            cout<<"Stack is empty\n";
      else
      {
            Node *Temp;
            Temp=Top;
       cout<<"\nElements of the Stack:\n\n";
            while(Temp!=NULL)
            {
                  cout<<Temp->Name<<endl;
                  cout<<Temp->Marks<<endl<<endl;
                  Temp=Temp->Next;
            }
      }
}

Stack::~Stack()                 //Destructor
{
      Node *Temp;
      while(Top!=NULL)
      {
            Temp=Top;
            Top=Top->Next;
            delete(Temp);
      }
}

void main()
{
      Stack S;
      int Choice;
      do
      {
            cout<<"1.PUSH\n2.POP\n3.DISPLAY\n4.QUIT\n";
            cin>>Choice;
            switch(Choice)
            {
                  case 1: S.Push();break;
                  case 2: S.Pop();break;
                  case 3: S.Display();break;
                  case 4: break;
                  default:cout<<"Wrong Choice";}
            }while(Choice!=4);
}
```


C:\TCWIN45\BIN\PRG19.EXE
4.QUIT
1
Enter Name and Marks of new node:
Mohan
200
1.PUSH
2.POP
3.DISPLAY
4.QUIT
3
Elements of the Stack:
Mohan
200
Aditya
100
1.PUSH
2.POP
3.DISPLAY
4.QUIT

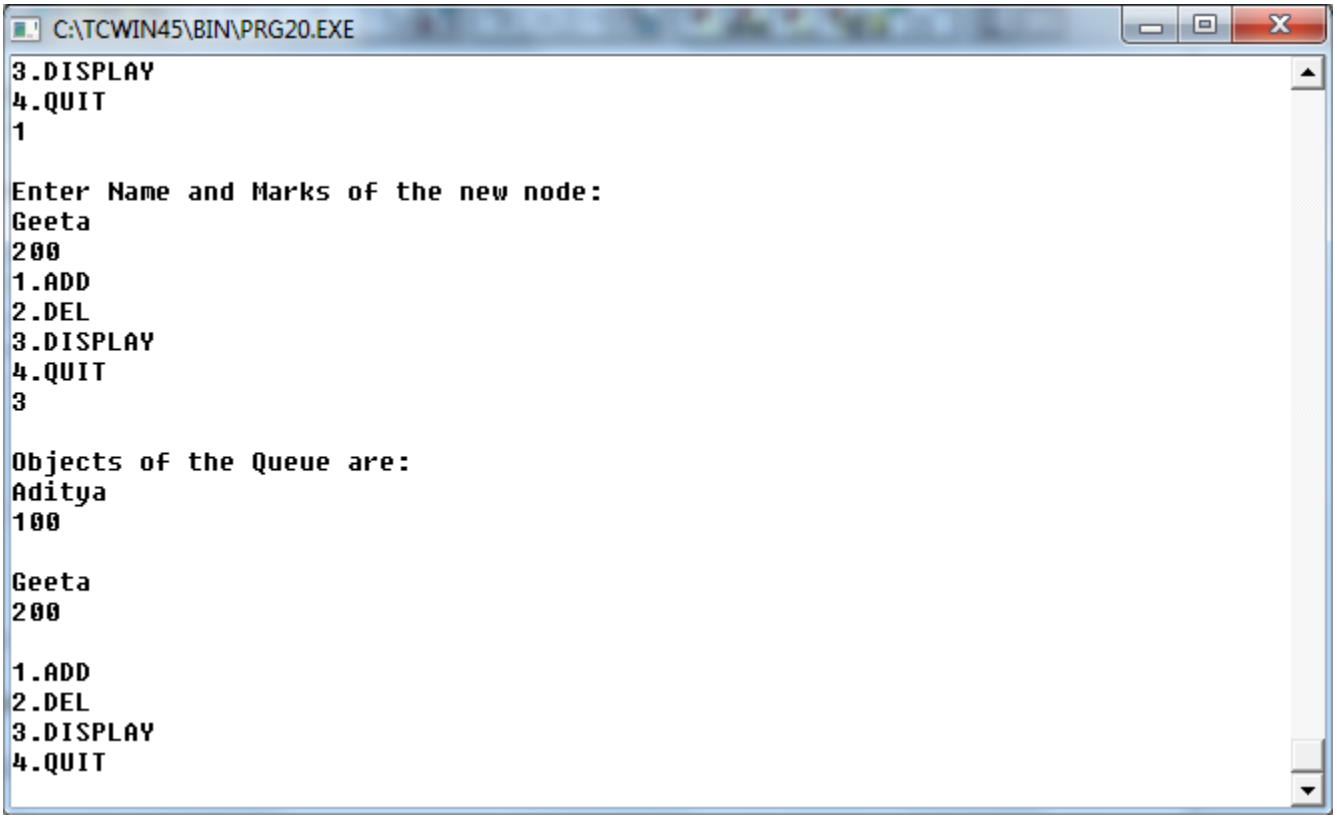

C:\TCWIN45\BIN\PRG19.EXE
100
1.PUSH
2.POP
3.DISPLAY
4.QUIT
2
One node deleted.
1.PUSH
2.POP
3.DISPLAY
4.QUIT
3
Elements of the Stack:
Aditya
100
1.PUSH
2.POP
3.DISPLAY
4.QUIT

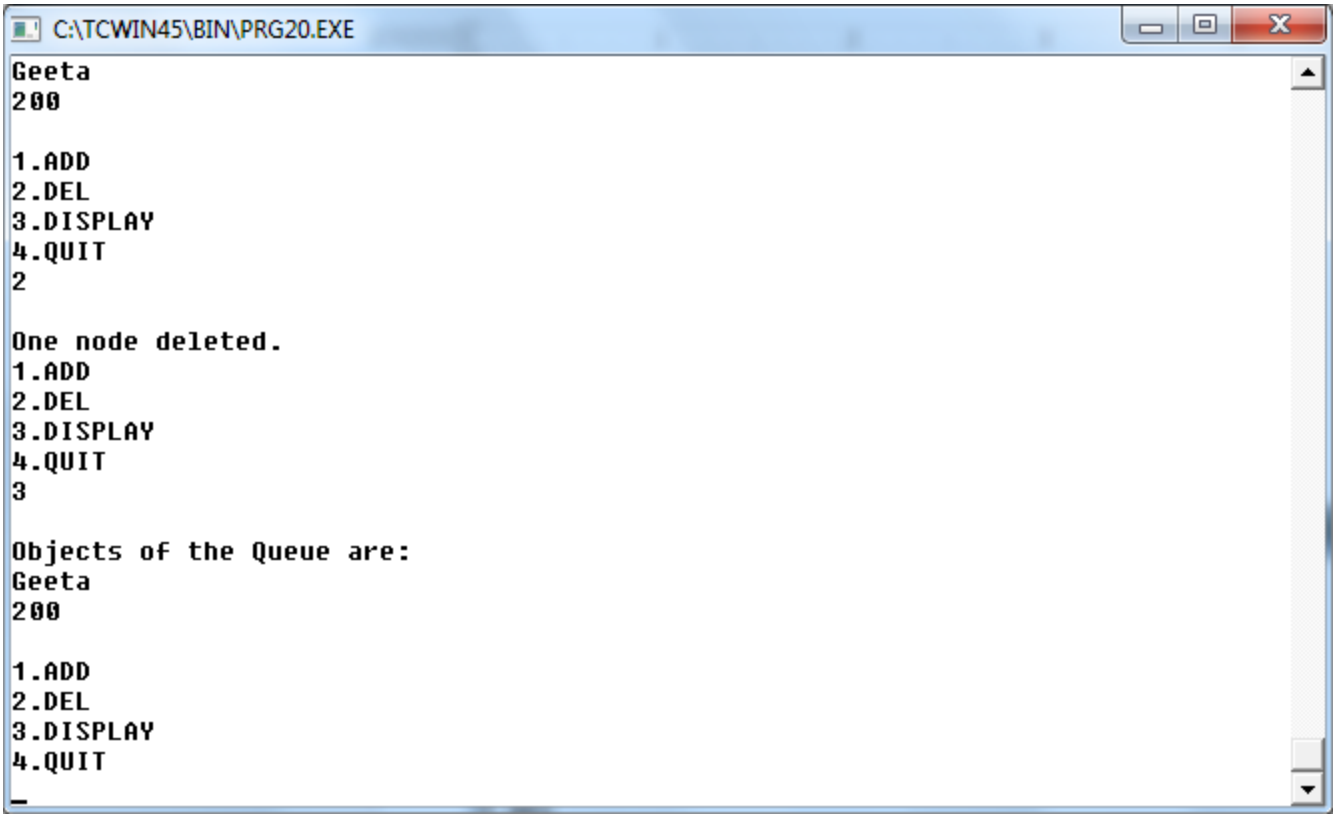
Program 20:
Linked list implementation of QUEUE, where each node consist of name and marks of a student.

```
//    Program No. 20



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

struct Node
{
      char Name[20];
      int Marks;
      Node *Next;
};

class Queue
{
      Node *Front;
      Node *Rear;
      public:
      Queue(){Front=Rear=NULL;}
      void AddQ();
      void DelQ();
      void DisplayQ();
      ~Queue();
};
void Queue::AddQ()              //Addition of a node at the rear
{
      Node *Temp;
      Temp=new Node;

      cout<<"\nEnter Name and Marks of the new node:\n";
      gets(Temp->Name);
      cin>>Temp->Marks;

      Temp->Next=NULL;

      if(Front!=NULL)
      {
            Rear->Next=Temp;
            Rear=Temp;
      }
      else
            Front=Rear=Temp;
}
```

```
void Queue::DelQ()            //Deletion of the node at the front
{
     if(Front==NULL)
     cout<<"Underflow!!Queue is empty\n";

     else
     {
          Node *Temp;
          Temp=Front;
          Front=Front->Next;
          delete(Temp);
          cout<<"\nOne node deleted.\n";
          if(Front==NULL)Rear=NULL;
     }
}

void Queue::DisplayQ()             //Displays the Queue
{
     if(Front==NULL)
          cout<<"Queue is empty\n";

     else
     {
          Node *Temp;
          Temp=Front;
          cout<<"\nObjects of the Queue are:\n";
          while(Temp!=NULL)
          {
               cout<<Temp->Name<<endl;
               cout<<Temp->Marks<<endl<<endl;
               Temp=Temp->Next;
          }
     }
}

Queue::~Queue()               //Destructor
{
     Node *Temp;

     while(Front!=NULL)
     {
          Temp=Front;
          Front=Front->Next;
          delete(Temp);
     }
}
```

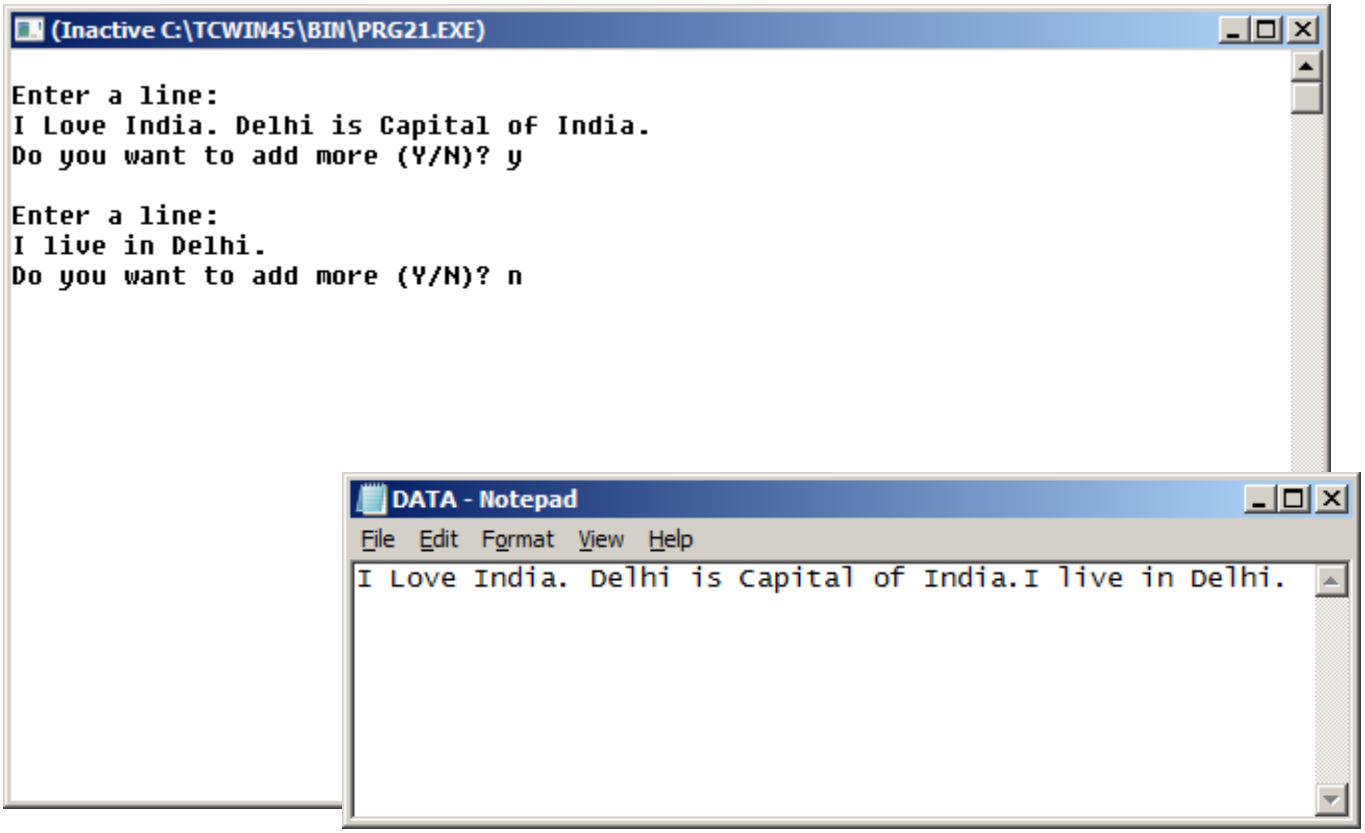
```
void main()
{
      Queue Q;
      int Ch;
      do
      {
            cout<<"1.ADD\n2.DEL\n3.DISPLAY\n4.QUIT\n";
            cin>>Ch;
            switch(Ch)
            {
                  case 1: Q.AddQ();break;
                  case 2: Q.DelQ();break;
                  case 3: Q.DisplayQ();break;
                  case 4: break;
            }
      }while(Ch!=4);
}
```




C:\TCWIN45\BIN\PRG20.EXE
Geeta
200
1.ADD
2.DEL
3.DISPLAY
4.QUIT
2
One node deleted.
1.ADD
2.DEL
3.DISPLAY
4.QUIT
3
Objects of the Queue are:
Geeta
200
1.ADD
2.DEL
3.DISPLAY
4.QUIT

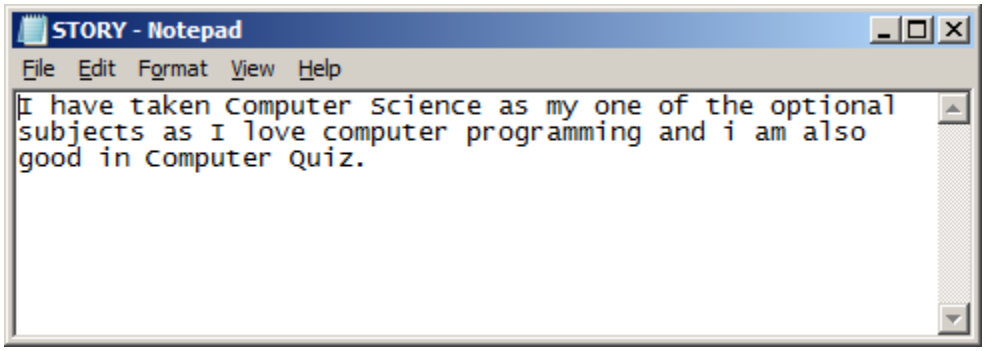
Program 21:
Create Text File: Read sentences through the keyboard and write those into a text file DATA.TXT

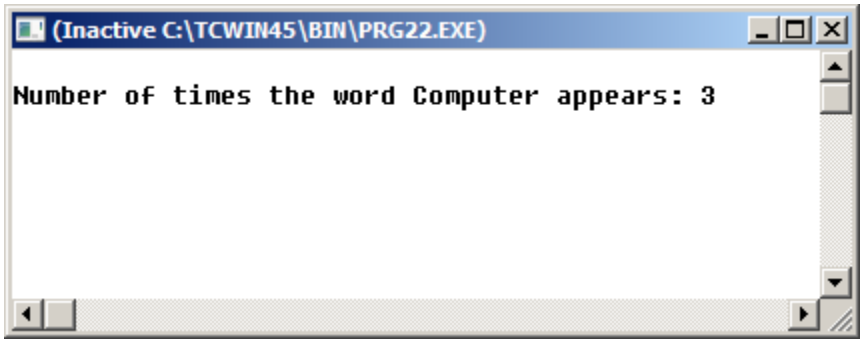
```
//    Program No. 21



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

void main()
{
      ofstream f1("DATA.TXT");
      char String[100];
      char Ans;
      do
      {
            cout<<"\nEnter a line:\n";
            gets(String);
            f1<<String;                  //Write the string into DATA.TXT
            cout<<"Do you want to add more (Y/N)? ";
            cin>>Ans;
      }while(Ans=='y' || Ans=='Y');
}
```



Program 22:
Read a text file "STORY.TXT" and count the number of times the word "computer" appears in it.

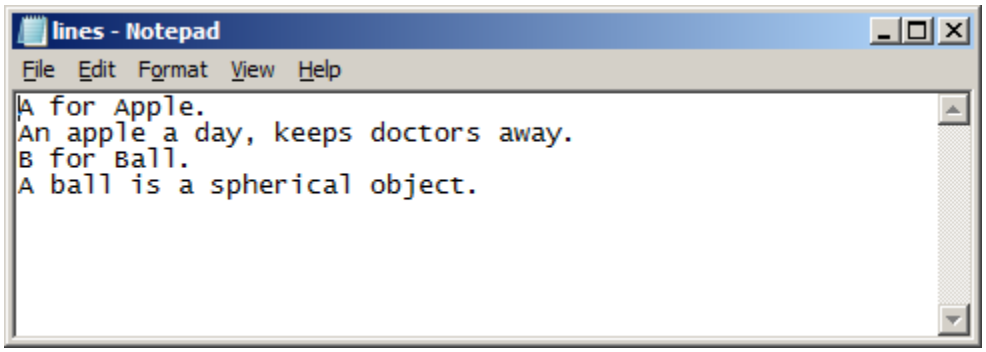
```
//    Program No. 22



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

void main()
{
      ifstream Fin("STORY.TXT");
      char Word[20];
      int Count=0;

      while(!Fin.eof())
      {
            Fin>>Word;

            if(strcmpi(Word,"Computer")==0)
                  Count++;  //Counts the number of times "Computer" appears
      }
      cout<<"\nNumber of times the word Computer appears: "<<Count<<endl;
}
```

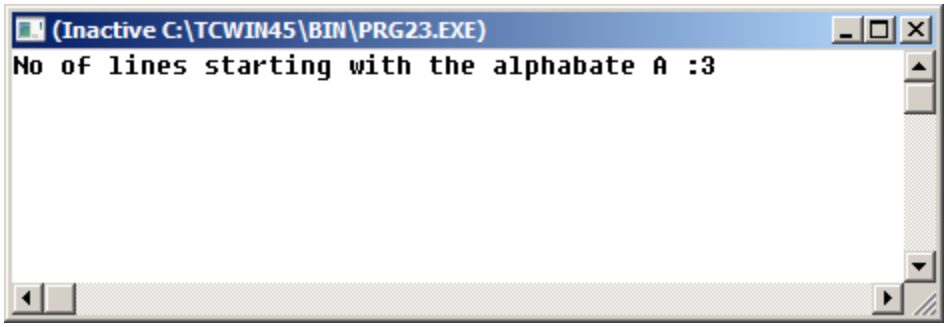




Program 23:
Read a Text File to count the number of lines starting with A.

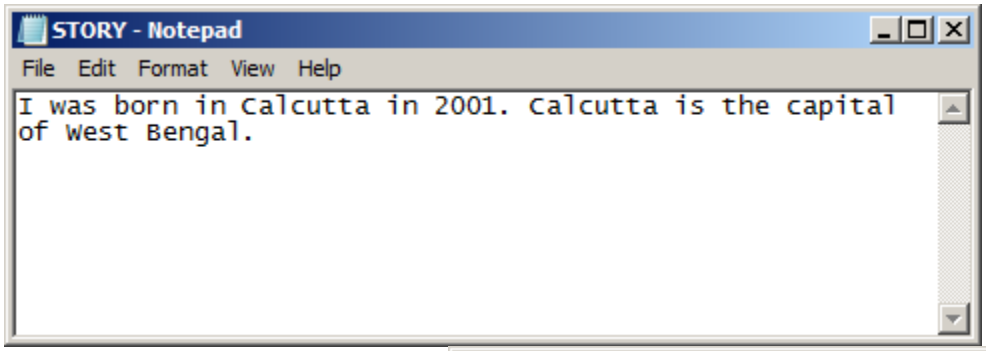
```
//    Program No. 23


#include<fstream.h>
void main()
{
      char Aline[80];
      int Count=0;
      ifstream Fin("LINES.TXT");

      while(!Fin.eof())
      {
            Fin.getline(Aline,80);

            if(Aline[0]=='a' || Aline[0]=='A')
                  Count++;
      }
      Fin.close();

      cout<<"No of lines starting with the alphabate A :"<<Count<<endl;
}
```

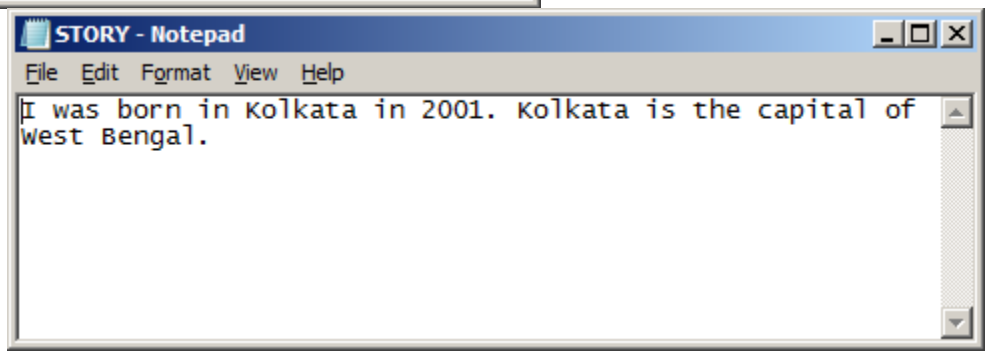




Program 24:
Modify a text file STORY.TXT by replacing all 'Calcutta' with 'Kolkata'.

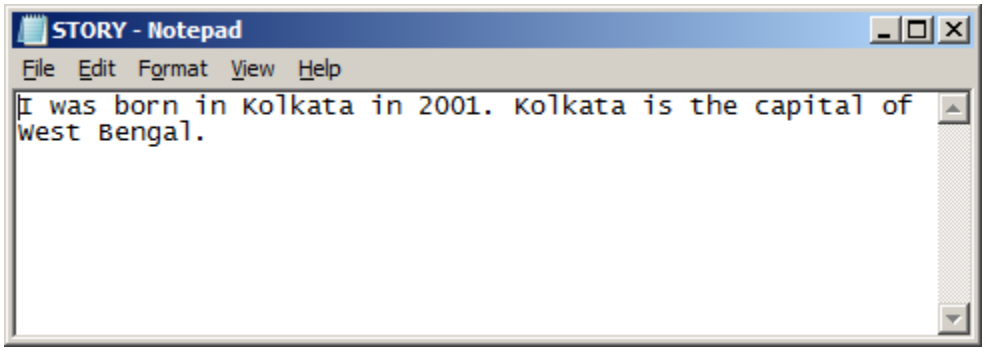
```
//    Program No. 24



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

void main()
{
      ifstream Fin("STORY.TXT");
      ofstream Fout("TEMP.TXT");
      char Word[20];
      while(!Fin.eof())
      {
            Fin>>Word;
            if(strcmpi(Word,"Calcutta")==0)     //Checks if Word is Calcutta
                  strcpy(Word,"Kolkata");       //Converts it into Kolkata
            Fout<<Word<<' ';
      }
      Fin.close();
      Fout.close();
      remove("STORY.TXT");
      rename("TEMP.TXT","STORY.TXT");
}
```

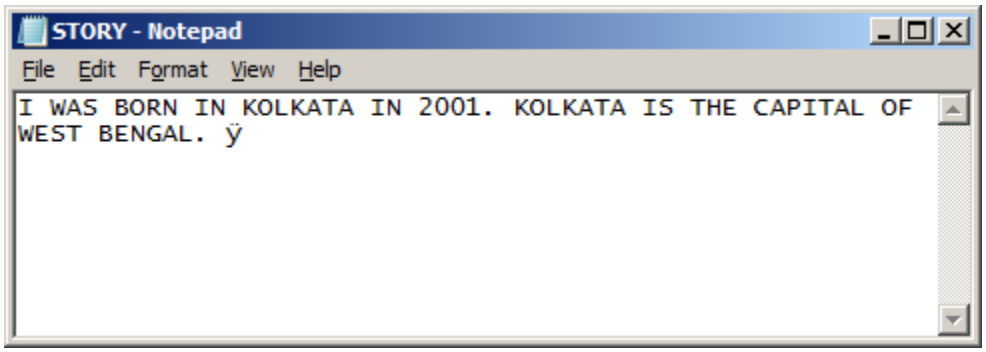




Program 25:
Modify a text file STORY.TXT by converting all characters into uppercase

```
//    Program No. 25



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

void main()
{
      ifstream Fin("STORY.TXT");
      ofstream Fout("TEMP.TXT");
      char Ch;
      while(!Fin.eof())
      {
            Fin.get(Ch);
            if(islower(Ch))                  //Checks if ch is lowercase
                  Ch=toupper(Ch);            //Converts it into uppercase
            Fout.put(Ch);
      }
      Fin.close();
      Fout.close();
      remove("STORY.TXT");
      rename("TEMP.TXT","STORY.TXT");
}
```

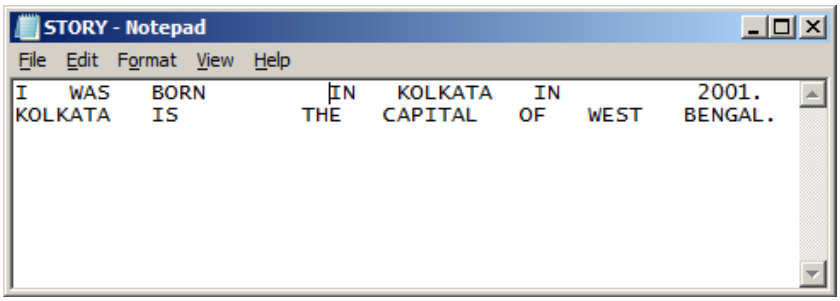


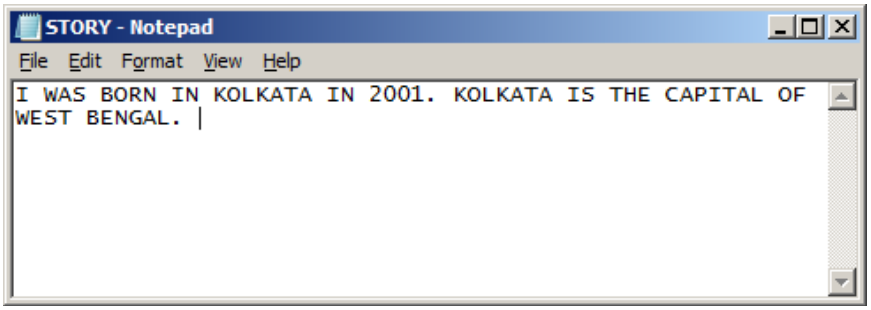


Program 26:

Modify a text file STORY.TXT 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.

```
//    Program No. 26



#include<fstream.h>
#include<stdio.h>
void main()
{
      ifstream Fin("STORY.TXT");
      ofstream Fout("TEMP.TXT");
      char Ch;
      int Space=0;
      while(!Fin.eof())
      {
            Fin.get(Ch);
            if(Ch==' ')
                  Space++;
             else
                  Space=0;
            if(Space<2)
                  Fout<<Ch;
      }
      Fin.close();
      Fout.close();
      remove("STORY.TXT");
      rename("TEMP.TXT","STORY.TXT");
}
```





Program 27:
Binary File - BANK
A menu based OOP to OPEN A NEW ACCOUNT, DISPLAY, WITHDRAW and DEPOSIT on some records stored in a binary file BANK.DAT. A record consists of Name (20 char) and Balance (Float). You can assume that the ACCOUNT NUMBER of a customer actually is the record number.

```
//   Program No. 27

#include<fstream.H>
#include<string.H>
#include<stdio.H>
#include<conio.H>

class Customer
{
     int   AccNo;
     char  Name[50];
     float Balance;
       public:
     void  Getdata(int);
     void  Showdata();
     int   RetAccNo();
     char *RetName();
     void  Transaction();
     void  Edit();
};

//   ************************ INPUT AN OBJECT ****************************
void Customer::Getdata(int AN)
{
     AccNo=AN;
     cout<<"\n\nEnter Name : "; gets(Name);
     cout<<"Enter Opening Balance : "; cin>>Balance;
}


//   ************************ DISPLAY AN OBJECT ****************************
void Customer::Showdata()
{
     cout<<"\n\nAccount Number : "<<AccNo<<endl;
     cout<<"Name : "<<Name<<endl;
     cout<<"Current Balance : "<<Balance<<endl<<endl;
}

//   ************** RETURN ACCOUNT NUMBER OF AN OBJECT ***************
int Customer::RetAccNo()
{
     return AccNo;
}
```

```
//    ********************* RETURN NAME OF AN OBJECT *********************
char *Customer::RetName()
{
     return Name;
}

//    ***************** WITHDRAW/DEPOSIT FOR AN OBJECT ******************
void Customer::Transaction()
{
     float amount;
     char ch;
     do{
          cout<<"\n\nCurrent Balance : "<<Balance<<endl;
          cout<<"\n\nPress W if you want to Withdraw  \n";
          cout<<"Press D if you want to Deposit \n";
          cout<<"Press Q if you want to Quit from this Menu\n";
          cout<<"Enter your option :";

          ch=getche();
          switch(ch)
          {
               case 'W':
               case 'w':      cout<<"\nEnter the amount to be Withdrawn :
";
                              cin>>amount;
                              if (Balance>amount)
                                   Balance -= amount;
                                    else
                                   cout<<"SORRY !! Insufficient Balance in the
Account !!\n";
                              cout<<"The Currenct Balance : "<<Balance<<endl;
                              break;
               case 'D':
               case 'd':      cout<<"\nEnter the amount to be Deposited :
";
                              cin>>amount;
                              if (amount>0)
                                   Balance += amount;
                                    else
                                   cout<<"SORRY !! Not a Valid Amount !!\n";
                              cout<<"The Currenct Balance : "<<Balance<<endl;
                              break;
               case 'Q':
               case 'q':      break;

               default : cout<<"\nWrong Option !! Try Again !!";
          }
     }while(ch!='Q' && ch!='q');
}
```

```
//   ******************** EDIT/ MODIFY AN OBJECT ************************
void Customer::Edit()
{
     char ch;
     do{
          cout<<"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 Current Balance \n";
          cout<<"Press C if you want no more change \n";
          cout<<"Enter your option :";
          ch=getche();
          switch(ch)
          {
               case 'A':
               case 'a':      cout<<"\nEnter new Name : ";
                              gets(Name);
                              break;
               case 'B':
               case 'b':      cout<<"\nEnter new Balance : ";
                              cin>>Balance;
                              break;
               case 'C':
               case 'c':      break;

               default : cout<<"\nWrong Option !! Try Again !!";
          }
     }while(ch!='C' && ch!='c');
}

//   **********DAY TO DAY TRANSACTION (DEPOSIT/WITHDRAW)***********
void Transaction()
{
     int san;
     Customer C;
     int Done = 0;

     cout<<"\n\nEnter the Account Number : ";
     cin>>san;


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

     while( f1.read((char*) &C, sizeof(C)) )
     {
          if(C.RetAccNo()==san)
          {
```

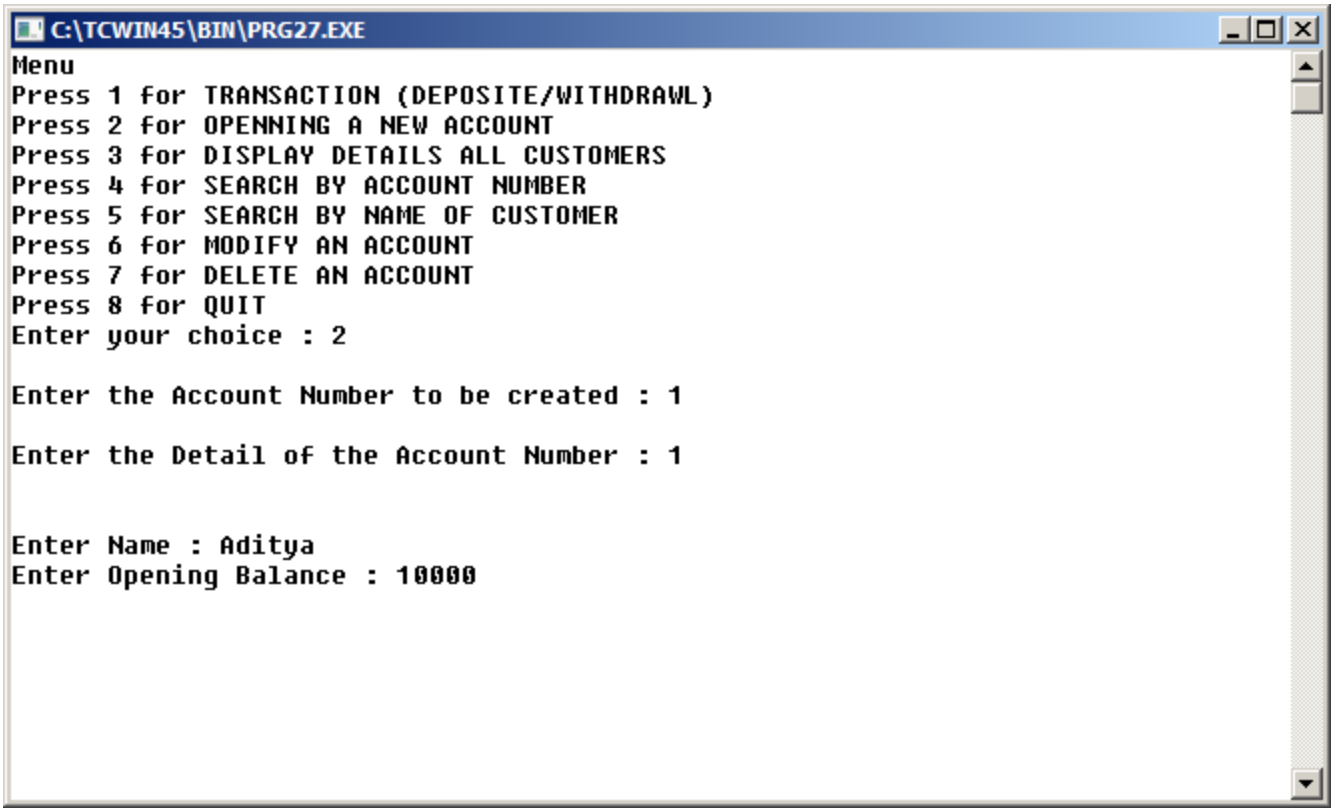
```
                Done++ ;
                C.Transaction();
            }
            f2.write((char *) &C, sizeof(C));
        }

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

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

//    ******************** APPEND (ADD) A NEW RECORD *********************
void Append()
{
        Customer C;
        int NewAccNo;
        int Found ;

        do{
            cout<<"\n\nEnter the Account Number to be created : ";
            cin>>NewAccNo;

            Found = 0;

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

            while( f1.read((char*) &C, sizeof(C)) )
            {
                if(C.RetAccNo()==NewAccNo)
                {
                    Found++ ;
                }
            }

            f1.close();

            if (Found != 0)
            cout<<"\n\n SORRY ! THIS ACCOUNT NUMBER ALREADY EXISTS!! \n\n";
        }while(Found!=0);

        cout<<"\nEnter the Detail of the Account Number : "<<NewAccNo<<endl;
        C.Getdata(NewAccNo);
```

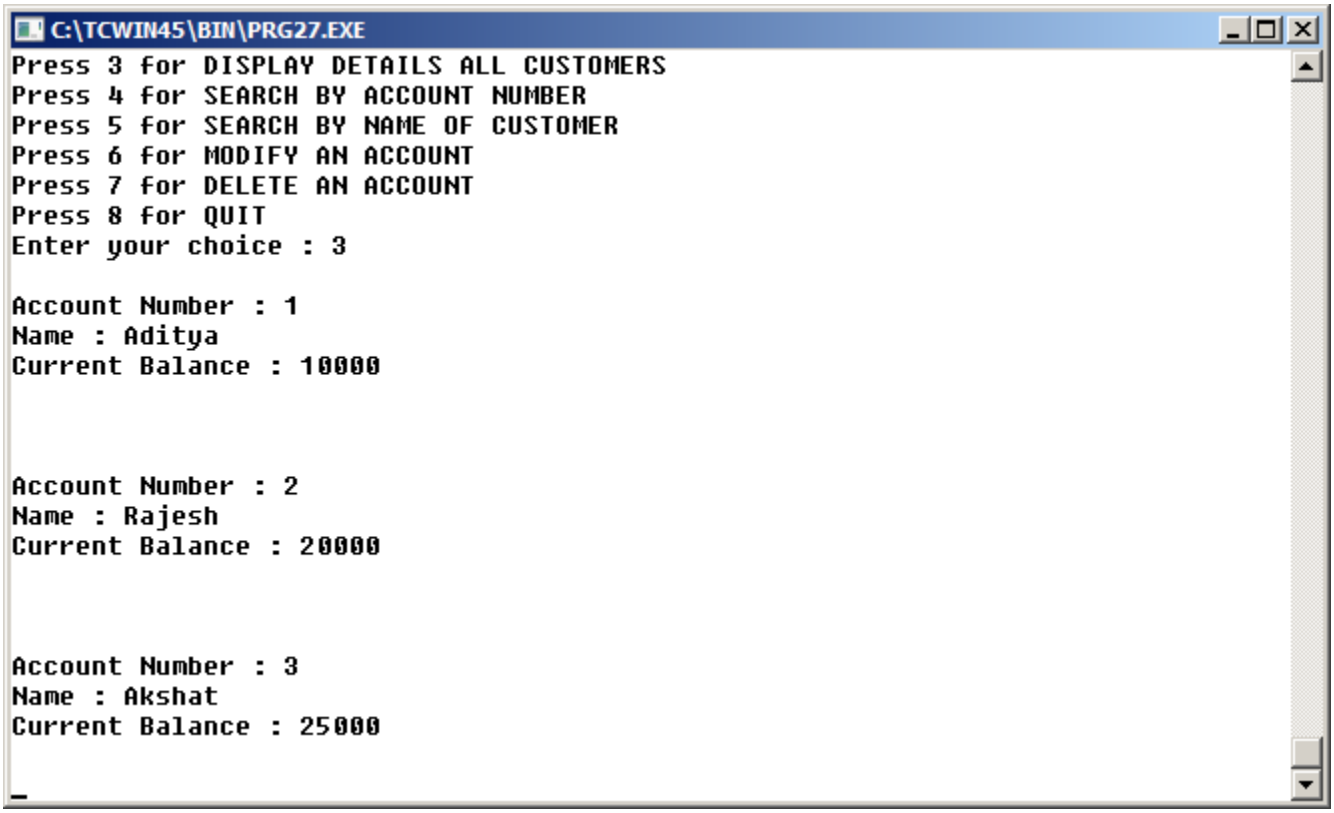
```
     fstream f2;
     f2.open("BANK.DAT",ios::binary|ios::app);
     f2.write((char*) &C, sizeof(C));
     f2.close();
}

//   ********************** DISPLAY ALL RECORDS ************************
void DisplayAll()
{
     Customer C;
     fstream f1;
     f1.open("BANK.DAT",ios::binary|ios::in);

     while( f1.read((char*) &C, sizeof(C)) )
     {
          C.Showdata();
     }

     f1.close();
}

//   ************** SEARCH A RECORD BY ACCOUNT NUMBER ****************
void Search1()
{
     int san;
     Customer C;
     int Found = 0;

     cout<<"\n\nEnter the Account Number to be searched : ";
     cin>>san;

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

     while( f1.read((char*) &C, sizeof(C)) )
     {
          if(C.RetAccNo()==san)
          {
               Found++ ;
               C.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";
```

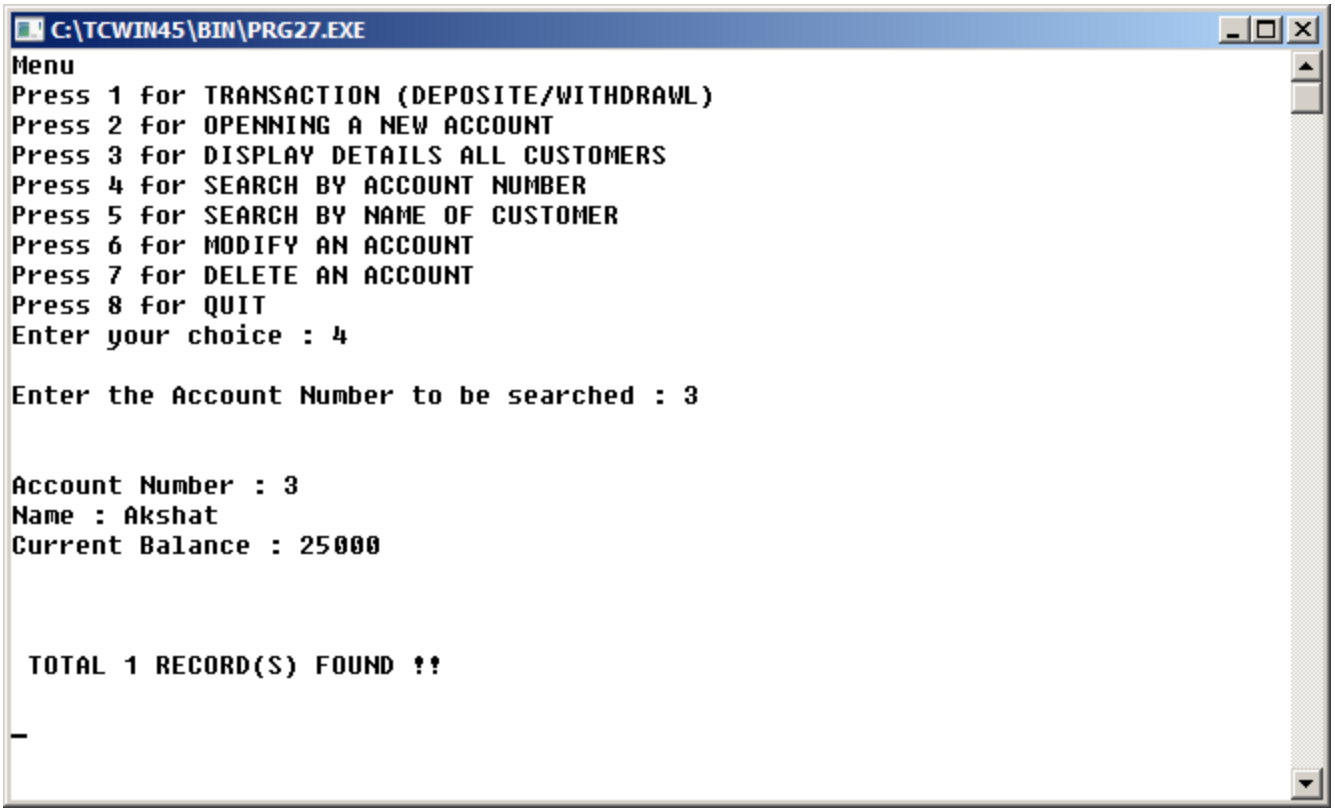
```
}

//   ******************** SEARCH A RECORD BY NAME **********************
void Search2()
{
     char sn[50];
     Customer C;
     int Found = 0;

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

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

     while( f1.read((char*) &C, sizeof(C)) )
     {
          if(strcmp(C.RetName(), sn)==0)
          {
               Found++ ;
               C.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";
}

//   ********************** EDIT/ MODIFY A RECORD ************************
void Modify()
{
     int san;
     Customer C;
     int Modified = 0;

     cout<<"\n\nEnter the Account Number to be modified : ";
     cin>>san;

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

     while( f1.read((char*) &C, sizeof(C)) )
     {
          if(C.RetAccNo()==san)
```

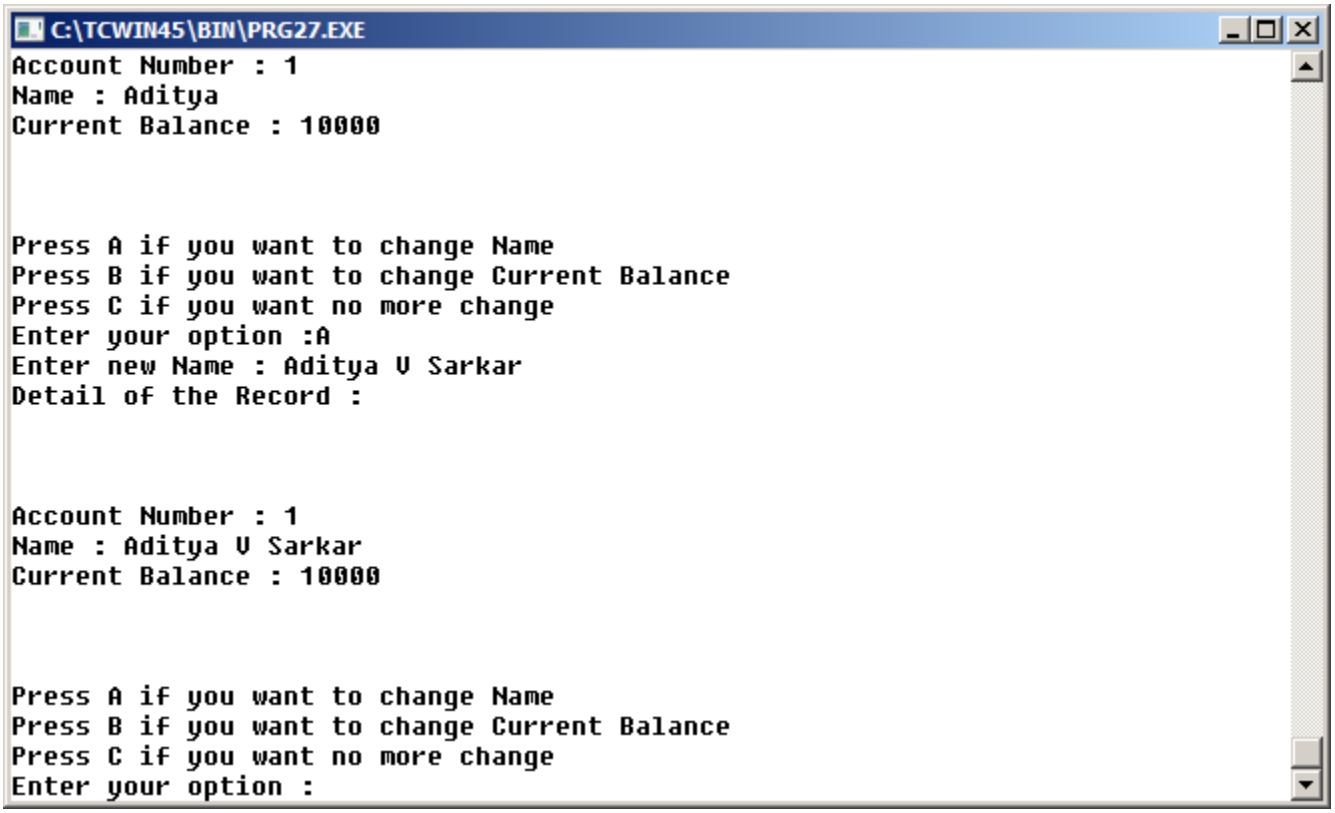
```
            {
                  Modified++ ;
                  C.Edit();
            }
            f2.write((char *) &C, sizeof(C));
      }

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

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

//    ************************ DELETE A RECORD ***************************
void Delete()
{
      int san;
      Customer C;
      int Deleted = 0;

      cout<<"\n\nEnter the Account Number to be deleted : ";
      cin>>san;


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

      while( f1.read((char*) &C, sizeof(C)) )
      {
            if(C.RetAccNo()==san)
                  Deleted ++ ;
                    else
                  f2.write((char *) &C, sizeof(C));
      }

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

      if (Deleted == 0)
            cout<<"\n\n NO MATCH FOUND !! \n\n";
              else
            {
                  remove("BANK.DAT");
```

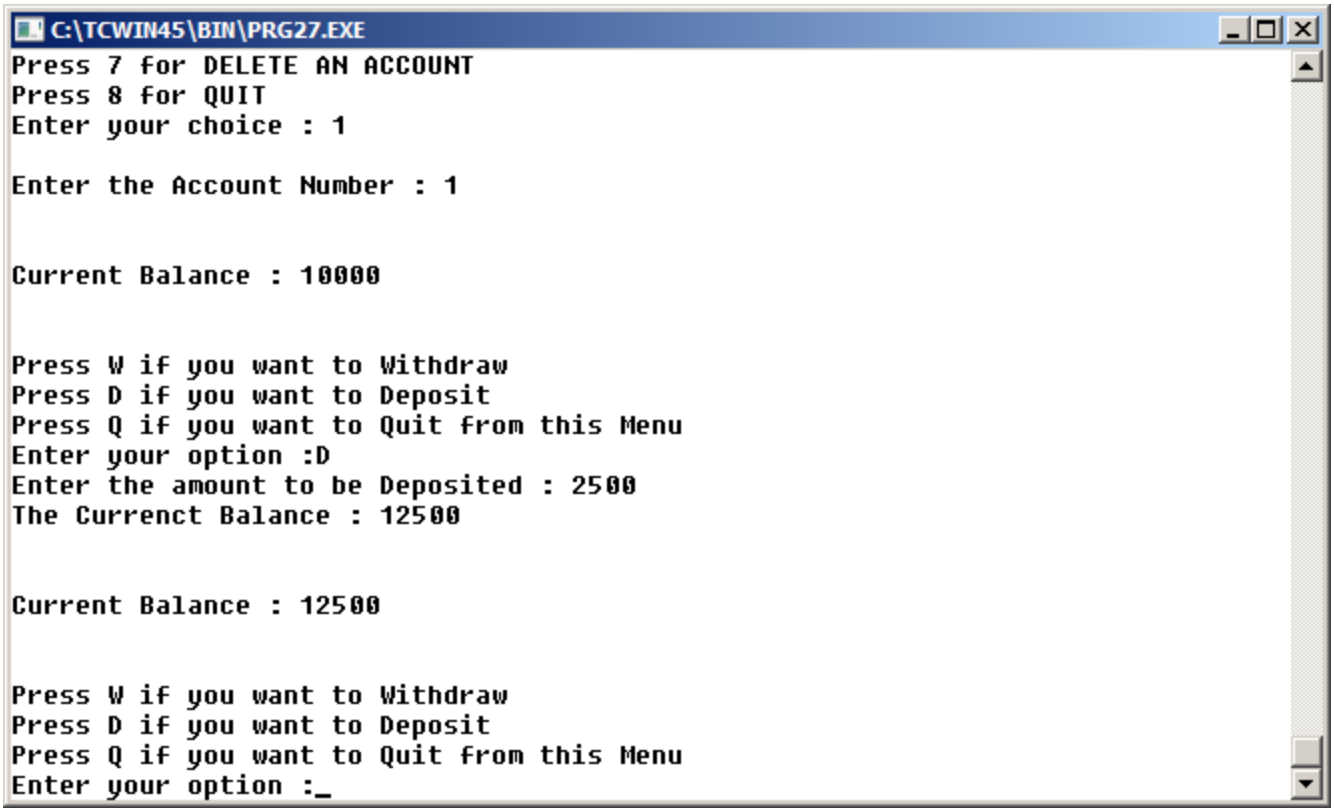
```
                rename("TEMP.DAT" , "BANK.DAT");
            }
}

void main()
{
     char choice;
     do{
          clrscr();
          cout<<"Menu\n";
          cout<<"Press 1 for TRANSACTION (DEPOSITE/WITHDRAWL) \n";
          cout<<"Press 2 for OPENNING A NEW ACCOUNT \n";
          cout<<"Press 3 for DISPLAY DETAILS ALL CUSTOMERS \n";
          cout<<"Press 4 for SEARCH BY ACCOUNT NUMBER \n";
          cout<<"Press 5 for SEARCH BY NAME OF CUSTOMER \n";
          cout<<"Press 6 for MODIFY AN ACCOUNT \n";
          cout<<"Press 7 for DELETE AN ACCOUNT \n";
          cout<<"Press 8 for QUIT \n";
          cout<<"Enter your choice : ";
          choice=getche();
          switch(choice)
          {
               case '1': Transaction();break;
               case '2': Append(); break;
               case '3': DisplayAll();  break;
               case '4': Search1();     break;
               case '5': Search2();     break;
               case '6': Modify(); break;
               case '7': Delete(); break;
               case '8':           break;
               default :cout<<"\nWrong Choice Entered !!\n\n";
          }
          getch();
     }while(choice!='8');
}
```

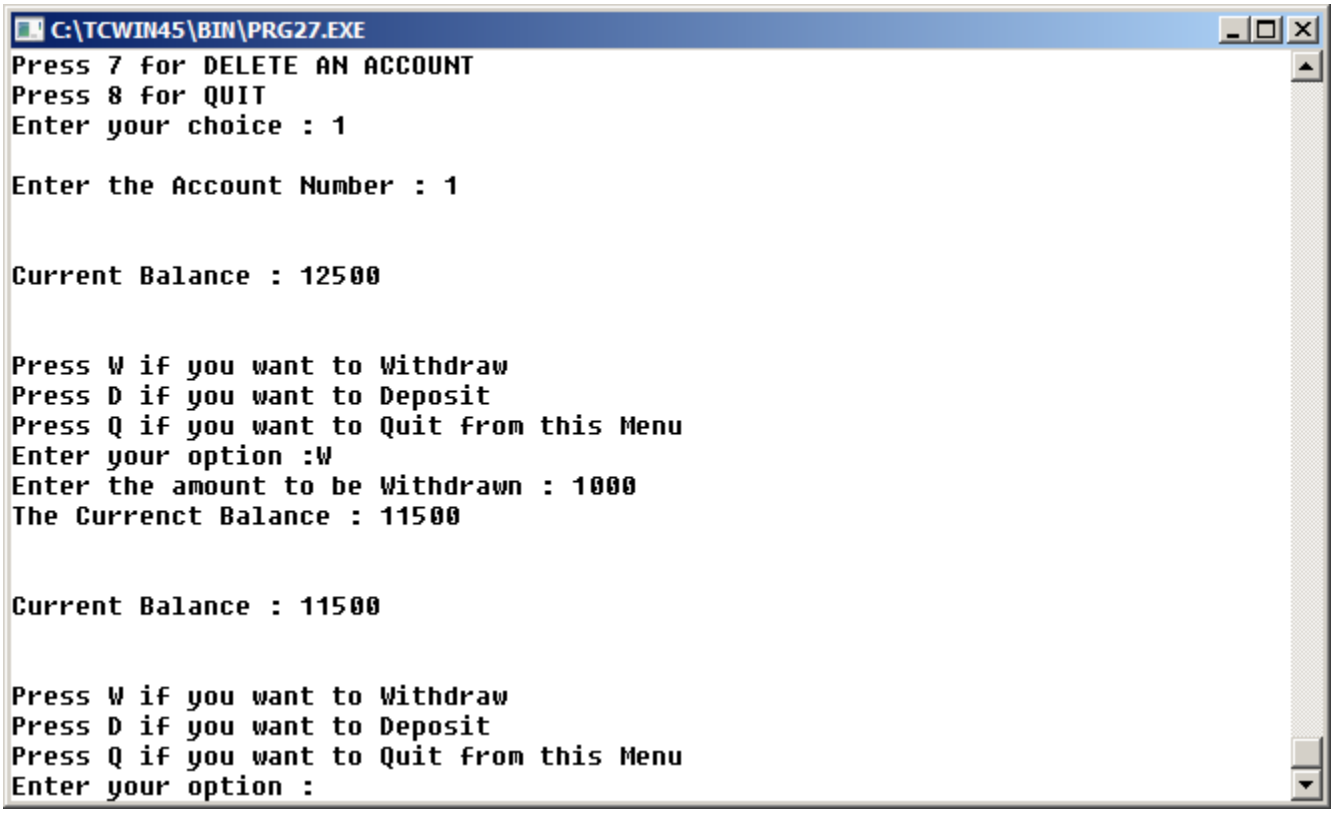

C:\TCWIN45\BIN\PRG27.EXE
Menu
Press 1 for TRANSACTION (DEPOSITE/WITHDRAWL)
Press 2 for OPENNING A NEW ACCOUNT
Press 3 for DISPLAY DETAILS ALL CUSTOMERS
Press 4 for SEARCH BY ACCOUNT NUMBER
Press 5 for SEARCH BY NAME OF CUSTOMER
Press 6 for MODIFY AN ACCOUNT
Press 7 for DELETE AN ACCOUNT
Press 8 for QUIT
Enter your choice : 2
Enter the Account Number to be created : 1
Enter the Detail of the Account Number : 1
Enter Name : Aditya
Enter Opening Balance : 10000


C:\TCWIN45\BIN\PRG27.EXE
Press 3 for DISPLAY DETAILS ALL CUSTOMERS
Press 4 for SEARCH BY ACCOUNT NUMBER
Press 5 for SEARCH BY NAME OF CUSTOMER
Press 6 for MODIFY AN ACCOUNT
Press 7 for DELETE AN ACCOUNT
Press 8 for QUIT
Enter your choice : 3
Account Number : 1
Name : Aditya
Current Balance : 10000
Account Number : 2
Name : Rajesh
Current Balance : 20000
Account Number : 3
Name : Akshat
Current Balance : 25000


C:\TCWIN45\BIN\PRG27.EXE
Menu
Press 1 for TRANSACTION (DEPOSITE/WITHDRAWL)
Press 2 for OPENNING A NEW ACCOUNT
Press 3 for DISPLAY DETAILS ALL CUSTOMERS
Press 4 for SEARCH BY ACCOUNT NUMBER
Press 5 for SEARCH BY NAME OF CUSTOMER
Press 6 for MODIFY AN ACCOUNT
Press 7 for DELETE AN ACCOUNT
Press 8 for QUIT
Enter your choice : 4
Enter the Account Number to be searched : 3
Account Number : 3
Name : Akshat
Current Balance : 25000
TOTAL 1 RECORD(S) FOUND !!


C:\TCWIN45\BIN\PRG27.EXE
Account Number : 1
Name : Aditya
Current Balance : 10000
Press A if you want to change Name
Press B if you want to change Current Balance
Press C if you want no more change
Enter your option :A
Enter new Name : Aditya V Sarkar
Detail of the Record :
Account Number : 1
Name : Aditya V Sarkar
Current Balance : 10000
Press A if you want to change Name
Press B if you want to change Current Balance
Press C if you want no more change
Enter your option :


C:\TCWIN45\BIN\PRG27.EXE
Press 7 for DELETE AN ACCOUNT
Press 8 for QUIT
Enter your choice : 1
Enter the Account Number : 1
Current Balance : 10000
Press W if you want to Withdraw
Press D if you want to Deposit
Press Q if you want to Quit from this Menu
Enter your option :D
Enter the amount to be Deposited : 2500
The Currenct Balance : 12500
Current Balance : 12500
Press W if you want to Withdraw
Press D if you want to Deposit
Press Q if you want to Quit from this Menu
Enter your option :_


C:\TCWIN45\BIN\PRG27.EXE
Press 7 for DELETE AN ACCOUNT
Press 8 for QUIT
Enter your choice : 1
Enter the Account Number : 1
Current Balance : 12500
Press W if you want to Withdraw
Press D if you want to Deposit
Press Q if you want to Quit from this Menu
Enter your option :W
Enter the amount to be Withdrawn : 1000
The Currenct Balance : 11500
Current Balance : 11500
Press W if you want to Withdraw
Press D if you want to Deposit
Press Q if you want to Quit from this Menu
Enter your option :

Program 28:
Binary File – ADDRESS BOOK
A menu based OOP to SEARCH, DISPLAY, ADD, DELETE and MODIFY some records stored in a binary file ADDRESS.DAT. A record consists of Name (20 char) and Telephone number and Email ID(char 30).

```
//    Program No. 28



#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  Check1(char nm[]);
      int  Check2(long tn);
      int  Check3(char em[]);
      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::Check1(char nm[])
```

```
{

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

//   ************ CHECK TELEPHONE NUMBER FOR AN OBJECT ***************
int Person::Check2(long tn)
{
     if (tn == Telno)
          return 1;
           else
          return 0;
}

//   ******************* CHECK EMAIL ID FOR AN OBJECT ********************
int Person::Check3(char em[])
{
     if (strcmp(em,Email) == 0)
          return 1;
           else
          return 0;
}

//   ******************** EDIT/ MODIFY AN OBJECT ************************
void Person::Edit()
{
     char ch;
     do{
          cout<<"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 :";
          ch=getche();
          switch(ch)
          {
               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<<"\nWrong Option !! Try Again !!";
          }
     }while(ch!='D' && ch!='d');
}

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

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

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

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

     f1.close();

}

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

     Person P;

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

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

     f1.close();
}

//   ******************** SEARCH A RECORD BY NAME **********************
void Search1()
{

     char sn[50];
```

```
      Person P;
      int Found = 0;

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

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

      while( f1.read((char*) &P, sizeof(P)) )
      {
            if(P.Check1(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 Search2()
{

      long stn;
      Person P;
      int Found = 0;

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

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

      while( f1.read((char*) &P, sizeof(P)) )
      {
            if(P.Check2(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 Search3()
{

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

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

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

	while( f1.read((char*) &P, sizeof(P)) )
	{
		if(P.Check3(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";

}

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

	char sn[50];
	Person P;
```

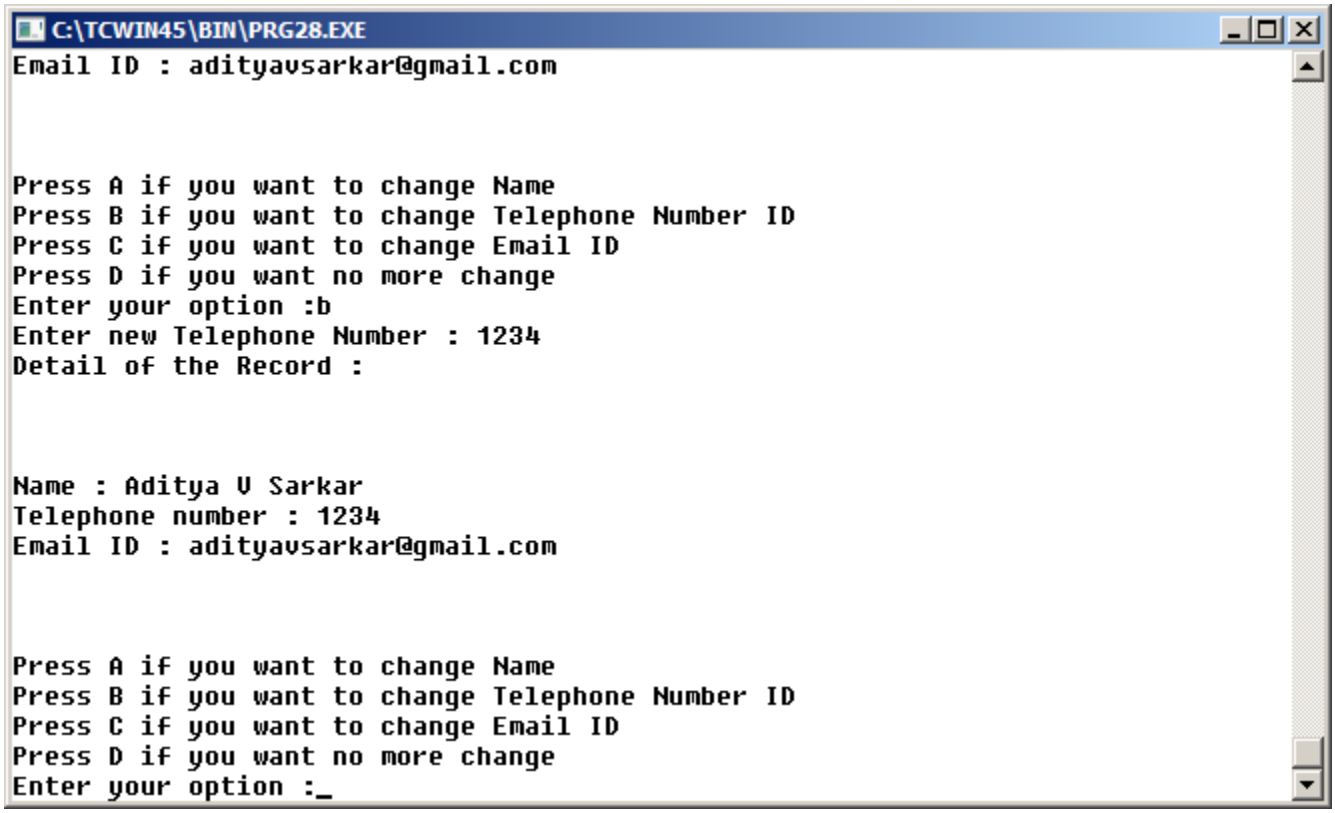
```
      int Modified = 0;

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

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

      while( f1.read((char*) &P, sizeof(P)) )
      {
           if(P.Check1(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("ADDRESS.DAT");
                rename("TEMP.DAT" , "ADDRESS.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("ADDRESS.DAT",ios::binary|ios::in);
      f2.open("TEMP.DAT",   ios::binary|ios::out);

      while( f1.read((char*) &P, sizeof(P)) )
      {
           if(P.Check1(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("ADDRESS.DAT");
                  rename("TEMP.DAT" , "ADDRESS.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 EMAIL ID \n";
             cout<<"Press 6 for MODIFY A RECORD \n";
             cout<<"Press 7 for DELETE A RECORD \n";
             cout<<"Press 8 for QUIT \n";
             cout<<"Enter your choice : ";
             choice=getch();
             switch(choice)
             {
                  case '1': Append(); break;
                  case '2': DisplayAll();  break;
                  case '3': Search1();     break;
                  case '4': Search2();     break;
                  case '5': Search3();     break;
                  case '6': Modify(); break;
                  case '7': Delete(); break;
                  case '8':           break;
                  default :cout<<"\nWrong Choice Entered !!\n\n";
             }
             getch();
        }while(choice!='8');
}
```


C:\TCWIN45\BIN\PRG28.EXE
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 EMAIL ID
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 for QUIT
Enter your choice :
Enter the Detail of the Record :
Enter Name : Aditya V Sarkar
Enter Telephone number : 9873653669
Enter Email ID : adityavsarkar@gmail.com


C:\TCWIN45\BIN\PRG28.EXE
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 EMAIL ID
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 for QUIT
Enter your choice :
Name : Aditya V Sarkar
Telephone number : 1234
Email ID : adityavsarkar@gmail.com
Name : Rupa Sarkar
Telephone number : 1220075984
Email ID : sarkarupa@gmail.com


C:\TCWIN45\BIN\PRG28.EXE
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 EMAIL ID
Press 6 for MODIFY A RECORD
Press 7 for DELETE A RECORD
Press 8 for QUIT
Enter your choice :
Enter the Name to be searched : Aditya V Sarkar
Name : Aditya V Sarkar
Telephone number : 1283719077
Email ID : adityavsarkar@gmail.com
TOTAL 1 RECORD(S) FOUND !!


C:\TCWIN45\BIN\PRG28.EXE
Email ID : adityavsarkar@gmail.com
Press A if you want to change Name
Press B if you want to change Telephone Number ID
Press C if you want to change Email ID
Press D if you want no more change
Enter your option :b
Enter new Telephone Number : 1234
Detail of the Record :
Name : Aditya V Sarkar
Telephone number : 1234
Email ID : adityavsarkar@gmail.com
Press A if you want to change Name
Press B if you want to change Telephone Number ID
Press C if you want to change Email ID
Press D if you want no more change
Enter your option :

Program 29:
Binary File – STUDENT
A menu based OOP to SEARCH, DISPLAY, ADD, DELETE and MODIFY some records of students stored in a binary file MARKS.DAT. A record consists of Name (50 char) and 5 marks.

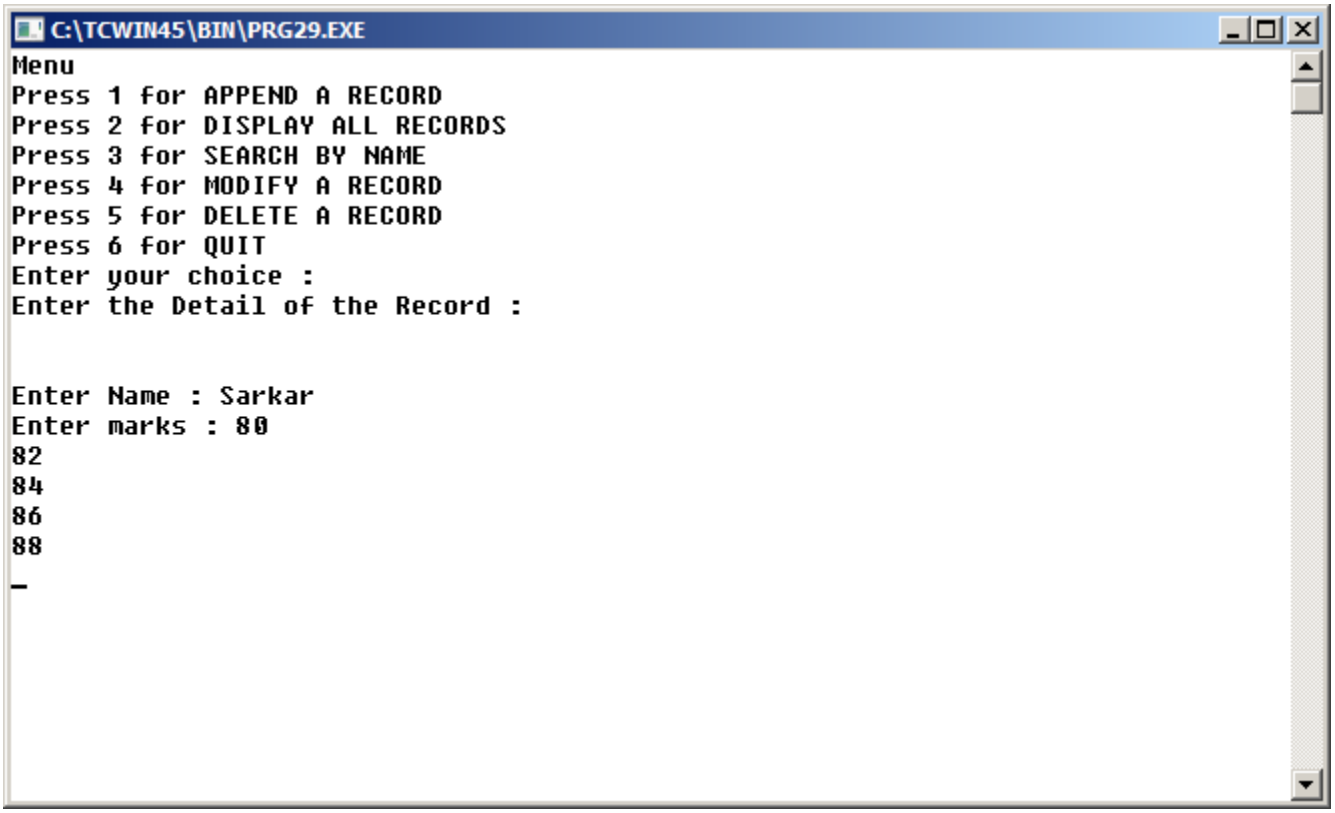
```
//    Program No. 29



#include<fstream.H>
#include<string.H>
#include<stdio.H>
#include<conio.H>


class Student
{
      char Name[50];
      int marks[5];

        public:
      void Getdata();
      void Showdata();
      int  Check1(char nm[]);
      int  Check2(int mr[]);

      void Edit();
};


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



//    ************************ DISPLAY AN OBJECT ***************************
void Student::Showdata()
{
      cout<<"\n\nName : "<<Name<<endl;
      for(int i=0;i<5;i++)
      cout<<"\nMarks:-"<<marks[i]<<endl;
}
```

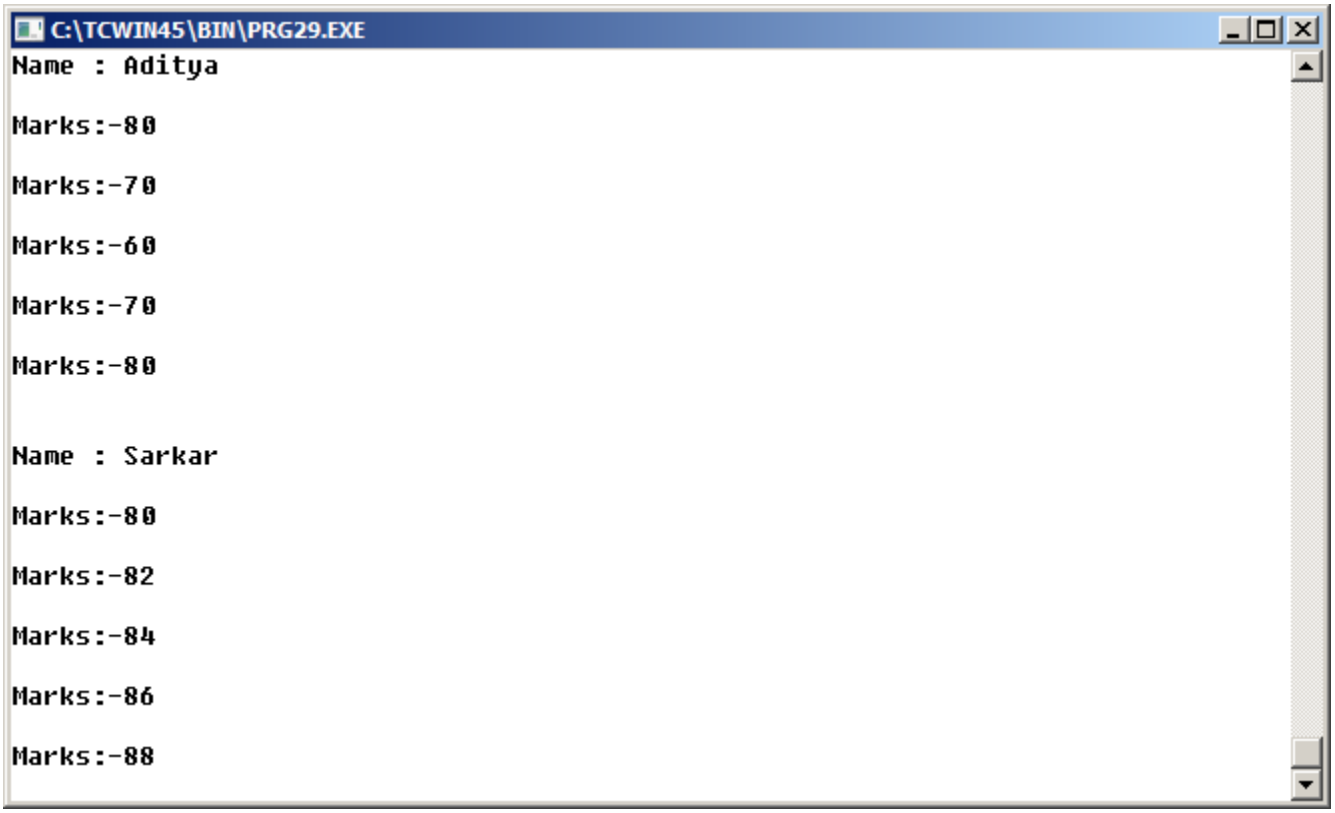
```
//   ******************** CHECK NAME FOR AN OBJECT **********************
int Student::Check1(char nm[])
{
     if (strcmp(nm,Name) == 0)
          return 1;
           else
          return 0;
}

//   ******************CHECK MARKS FOR AN OBJECT ******************
int Student::Check2(int mr[])
{
     for(int i=0;i<5;i++)
     {
          for(int j=0;j<5;j++)
          {
               if(mr[i]==marks[j])
               return 1;
          }
     }
   return 0;
}

//   ******************** EDIT/ MODIFY AN OBJECT ************************
void Student::Edit()
{
     char ch;
     int i;
     do{
          cout<<"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 marks \n";
          cout<<"Press C if you want no more change \n";
          cout<<"Enter your option :";
          ch=getche();

          switch(ch)
          {
               case 'A':
               case 'a':      cout<<"\nEnter new Name : ";
                              gets(Name);
                              break;
               case 'B':
               case 'b':      cout<<"\nEnter new marks : ";
                              for(i=0;i<5;i++)
                              {
                                   cin>>marks[i];
                              }
                              break;
```

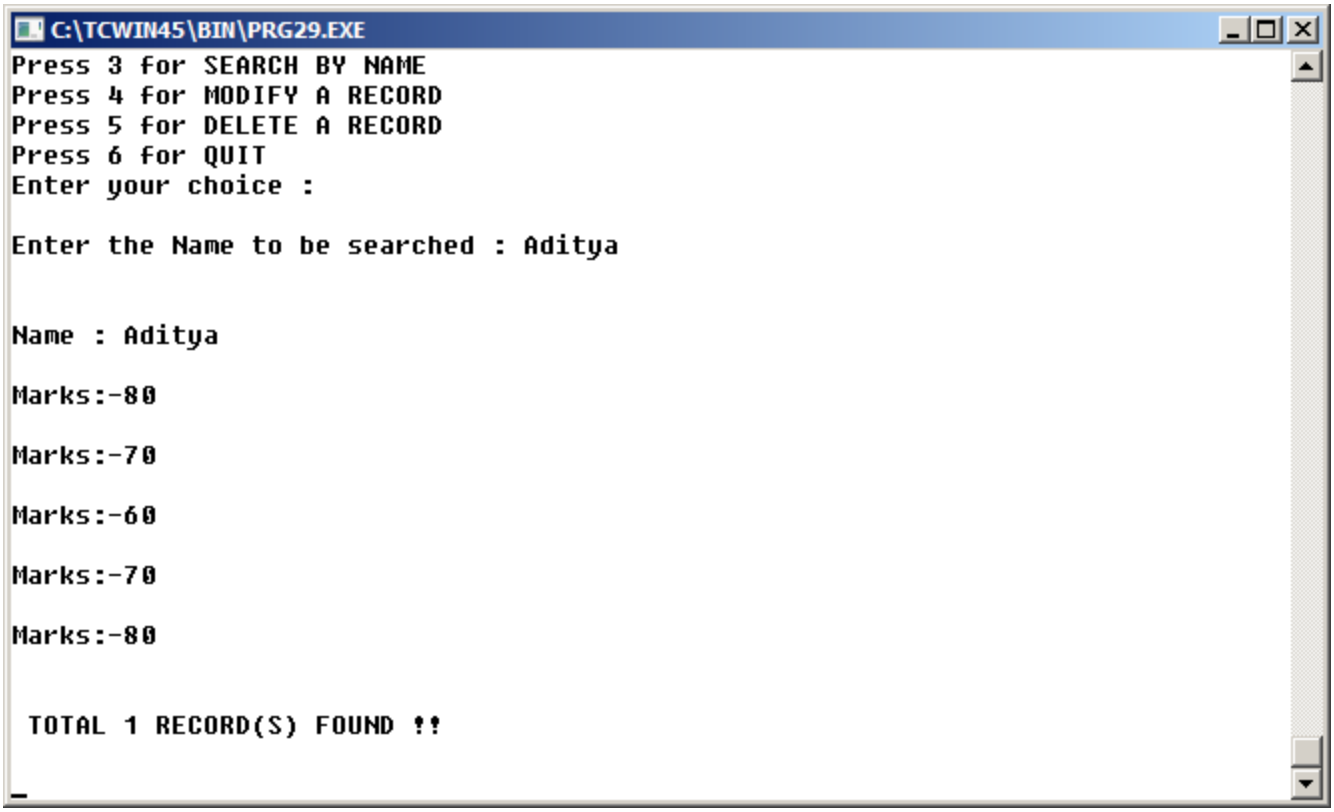
```
                case 'C':
                case 'c':      break;

                default : cout<<"\nWrong Option !! Try Again !!";
           }
     }while(ch!='D' && ch!='d');
}

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

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

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

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

     f1.close();

}

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

     Student P;

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

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

     f1.close();

}

//   *************** SEARCH A RECORD BY NAME ************************
void Search()
{
     char sn[50];
     Student P;
     int Found = 0;
     cout<<"\n\nEnter the Name to be searched : ";
```

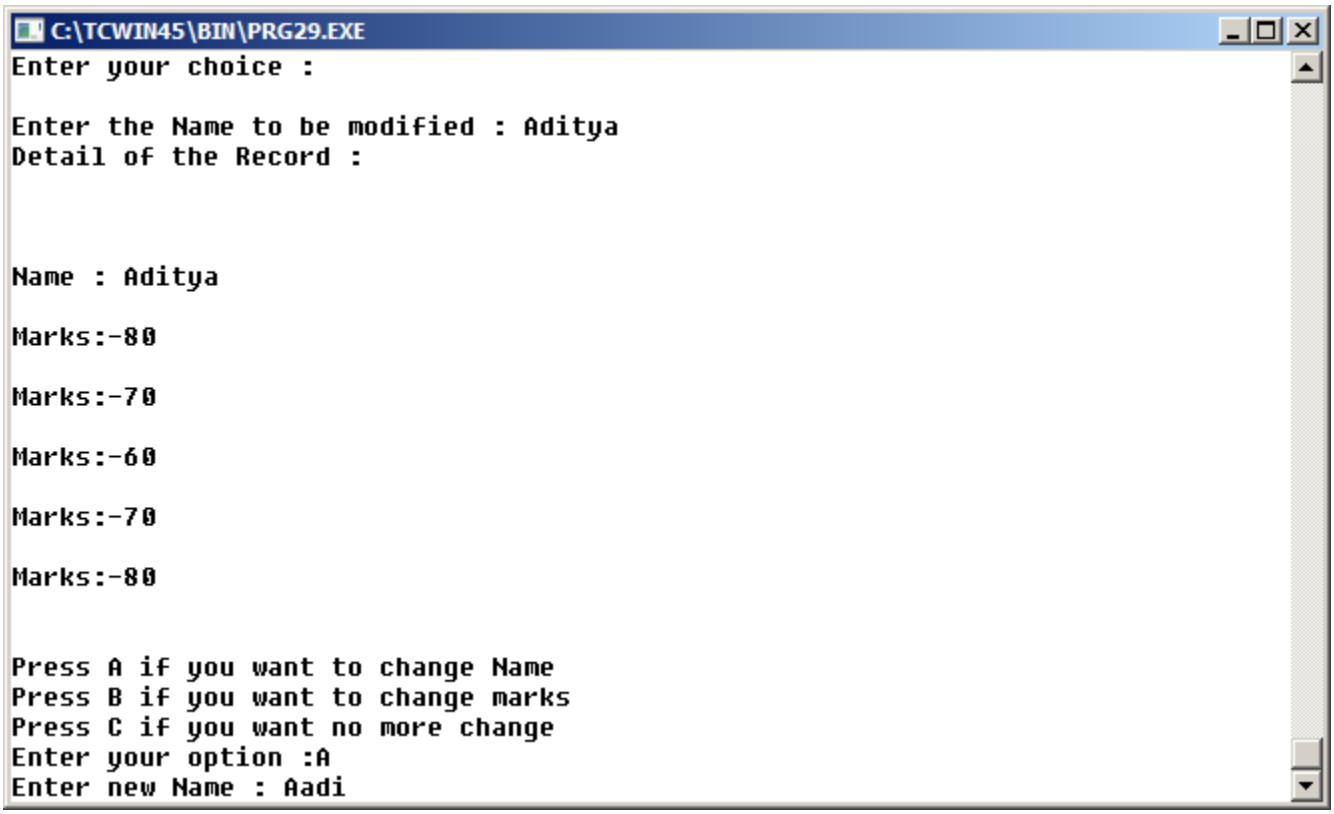
```
      gets(sn);
      fstream f1;
      f1.open("STUDENT.DAT",ios::binary|ios::in);
      while( f1.read((char*) &P, sizeof(P)) )
      {
            if(P.Check1(sn)==1)
            {
                  Found++ ;
                  P.Showdata();

            }
      }
      f1.close();
      if (Found == 0)
            cout<<"\n\n NO MATCH FOUND !! \n\n";
              else
            cout<<"\n\n TOTAL "<< Found << " RECORD(S) FOUND !! \n\n";
}

//    ********************** EDIT/ MODIFY A RECORD ************************
void Modify()
{
      char sn[50];
      Student P;
      int Modified = 0;
      cout<<"\n\nEnter the Name to be modified : ";
      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*) &P, sizeof(P)) )
      {
            if(P.Check1(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("STUDENT.DAT");
                  rename("TEMP.DAT" , "STUDENT.DAT");
            }
}
```

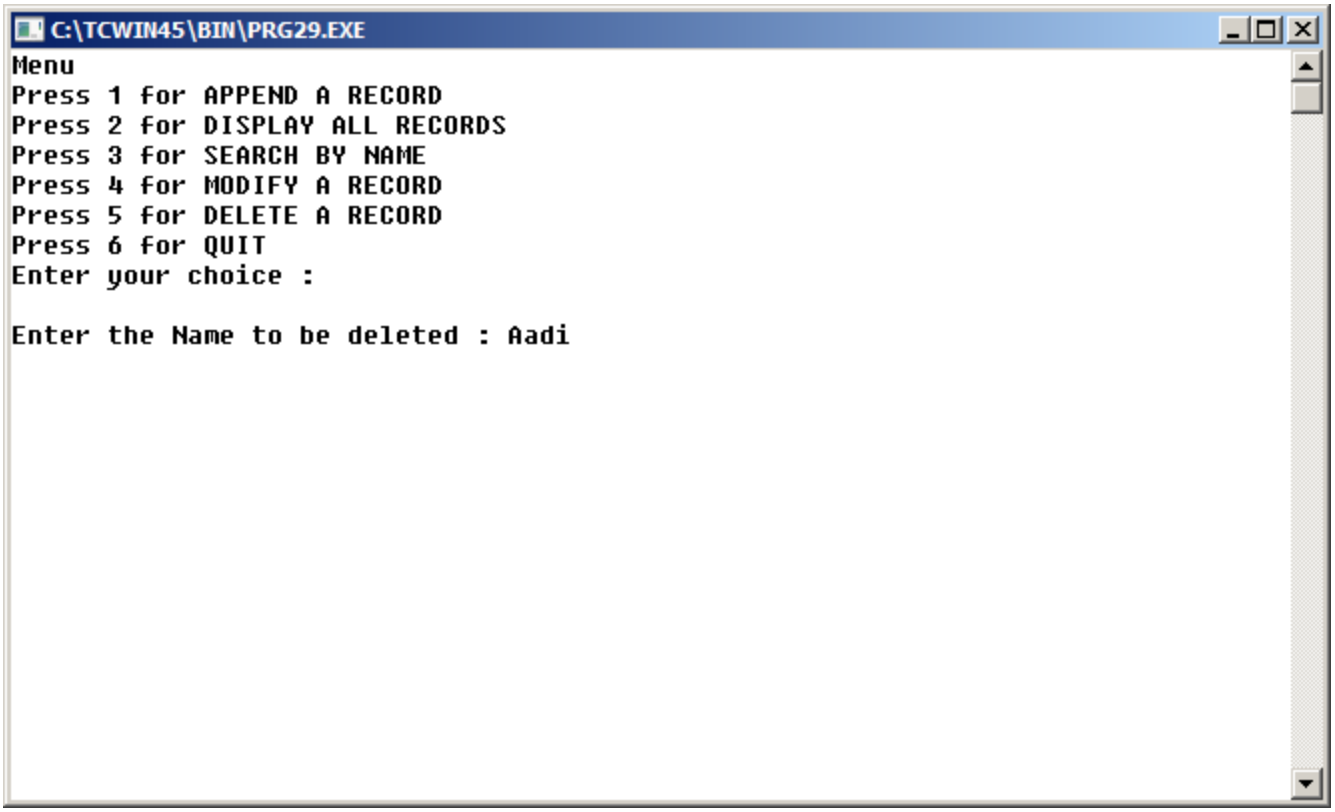
```
//   ************************ DELETE A RECORD ***************************
void Delete()
{    char sn[50];
     Student P;
     int Deleted = 0;
     cout<<"\n\nEnter the Name to be deleted : ";
     gets(sn);
     fstream f1,f2;
     f1.open("STUDENT.DAT",ios::binary|ios::in);
     f2.open("TEMP.DAT",   ios::binary|ios::out);
     while( f1.read((char*) &P, sizeof(P)) )
     {    if(P.Check1(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("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 MODIFY A RECORD \n";
          cout<<"Press 5 for DELETE A RECORD \n";
          cout<<"Press 6 for QUIT \n";
          cout<<"Enter your choice : ";
          choice=getch();
          switch(choice)
          {    case '1': Append(); break;
               case '2': DisplayAll();  break;
               case '3': Search(); break;

               case '4': Modify(); break;
               case '5': Delete(); break;
               case '6':           break;
               default :cout<<"\nWrong Choice Entered !!\n\n";
          }
     }while(choice!='6');
}
```

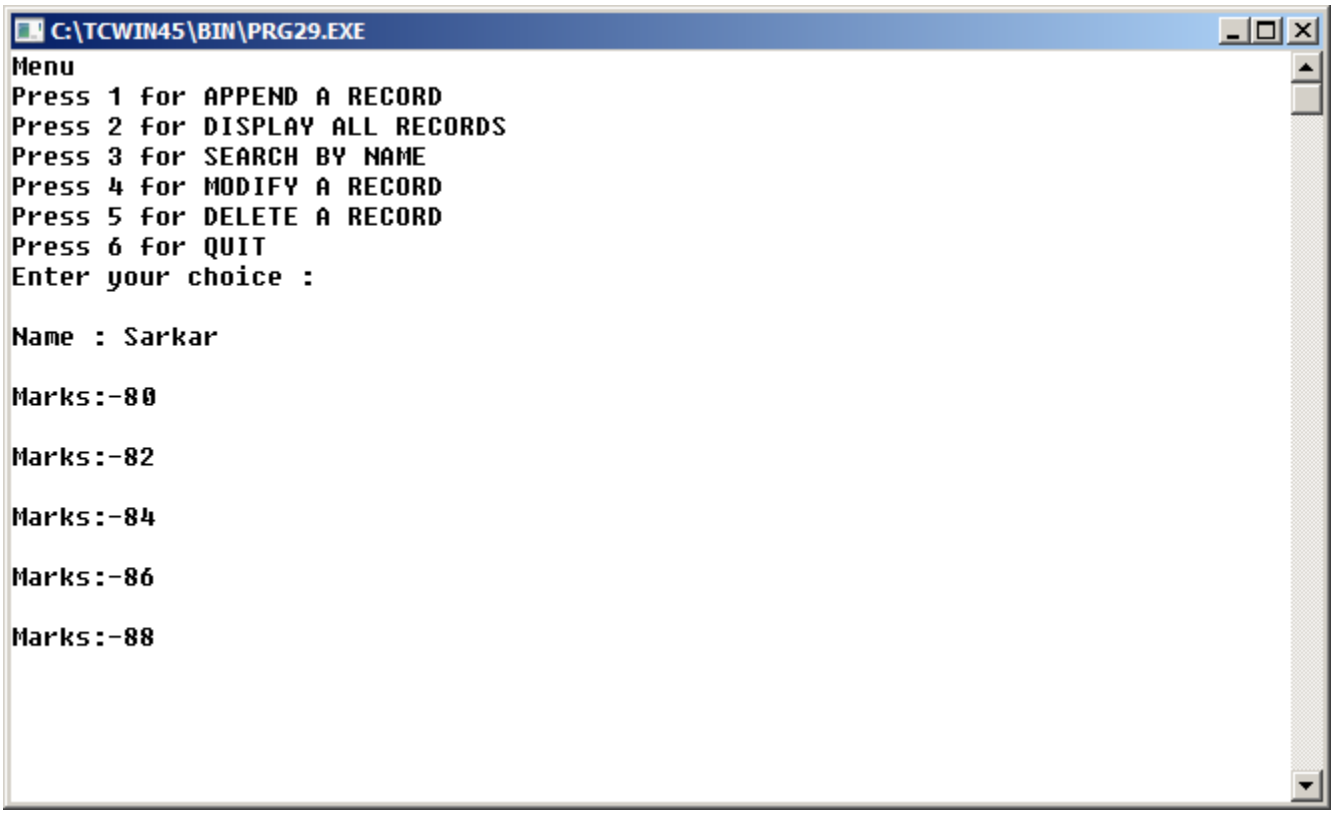

C:\TCWIN45\BIN\PRG29.EXE
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for MODIFY A RECORD
Press 5 for DELETE A RECORD
Press 6 for QUIT
Enter your choice :
Enter the Detail of the Record :
Enter Name : Sarkar
Enter marks : 80
82
84
86
88


C:\TCWIN45\BIN\PRG29.EXE
Name : Aditya
Marks:-80
Marks:-70
Marks:-60
Marks:-70
Marks:-80
Name : Sarkar
Marks:-80
Marks:-82
Marks:-84
Marks:-86
Marks:-88


C:\TCWIN45\BIN\PRG29.EXE
Press 3 for SEARCH BY NAME
Press 4 for MODIFY A RECORD
Press 5 for DELETE A RECORD
Press 6 for QUIT
Enter your choice :
Enter the Name to be searched : Aditya
Name : Aditya
Marks:-80
Marks:-70
Marks:-60
Marks:-70
Marks:-80
TOTAL 1 RECORD(S) FOUND !!


C:\TCWIN45\BIN\PRG29.EXE
Enter your choice :
Enter the Name to be modified : Aditya
Detail of the Record :
Name : Aditya
Marks:-80
Marks:-70
Marks:-60
Marks:-70
Marks:-80
Press A if you want to change Name
Press B if you want to change marks
Press C if you want no more change
Enter your option :A
Enter new Name : Aadi


C:\TCWIN45\BIN\PRG29.EXE
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for MODIFY A RECORD
Press 5 for DELETE A RECORD
Press 6 for QUIT
Enter your choice :
Enter the Name to be deleted : Aadi

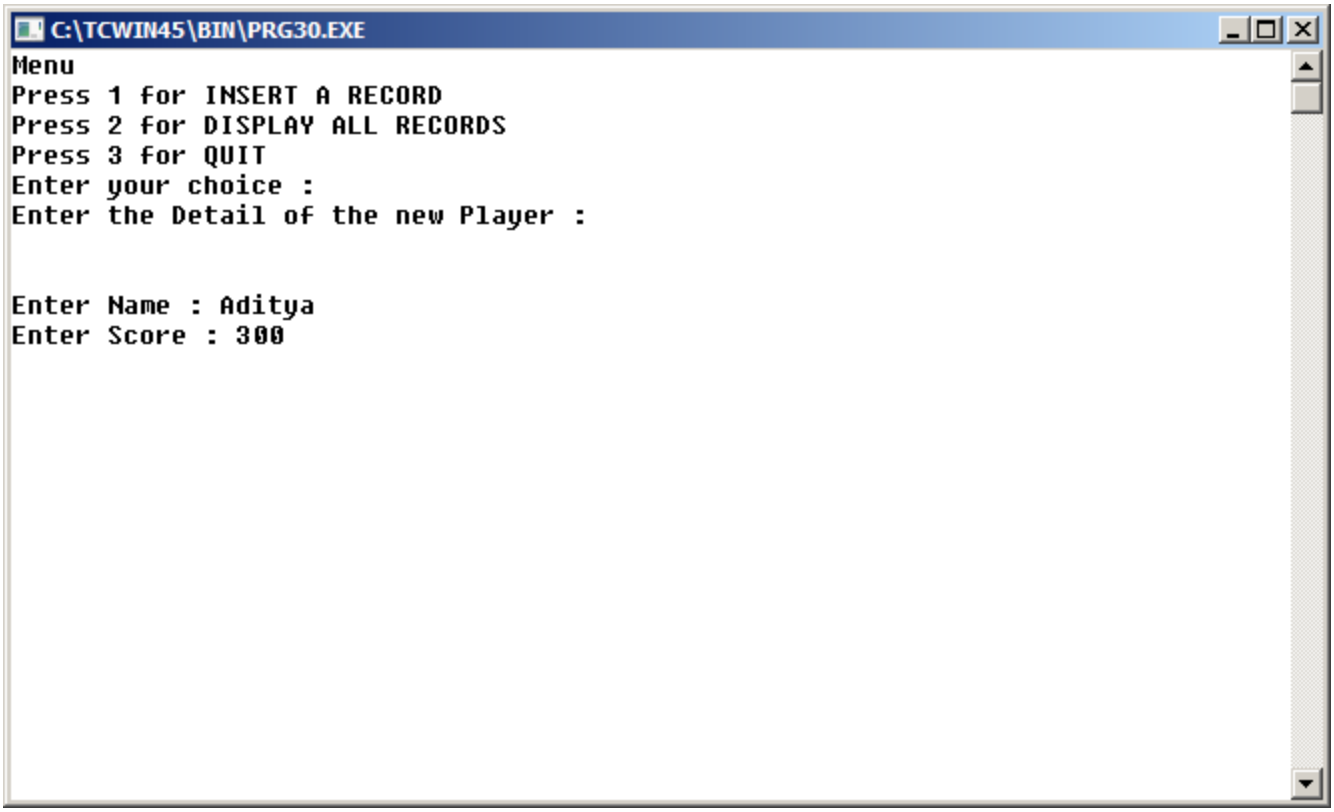

C:\TCWIN45\BIN\PRG29.EXE
Menu
Press 1 for APPEND A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for SEARCH BY NAME
Press 4 for MODIFY A RECORD
Press 5 for DELETE A RECORD
Press 6 for QUIT
Enter your choice :
Name : Sarkar
Marks:-80
Marks:-82
Marks:-84
Marks:-86
Marks:-88

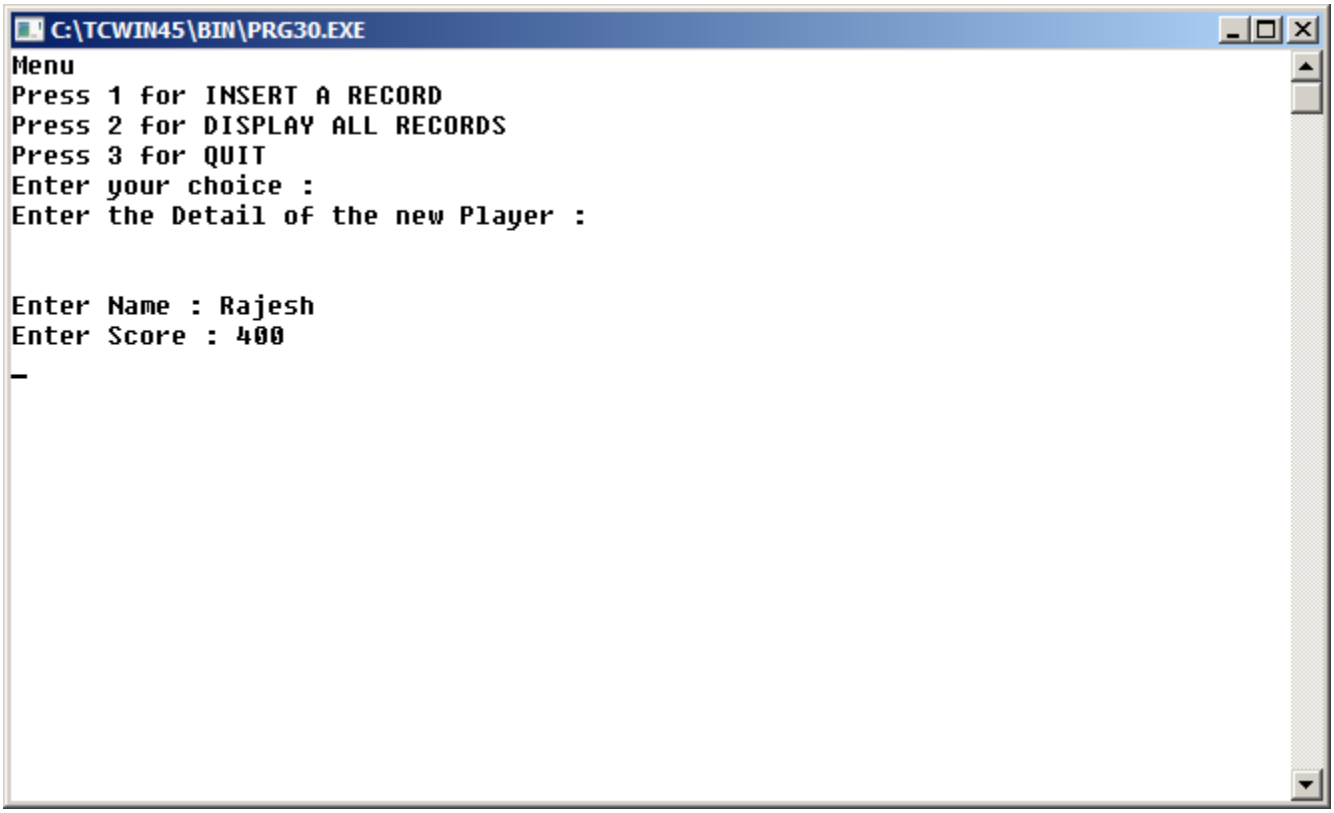
Program 30:
Binary File – HALL OF FAME
A menu based OOP to APPEND & DISPLAY some records of Player stored in a binary file HOF.DAT. A record consists of Name and Score. All records should be stored in the decreasing order of their scores.

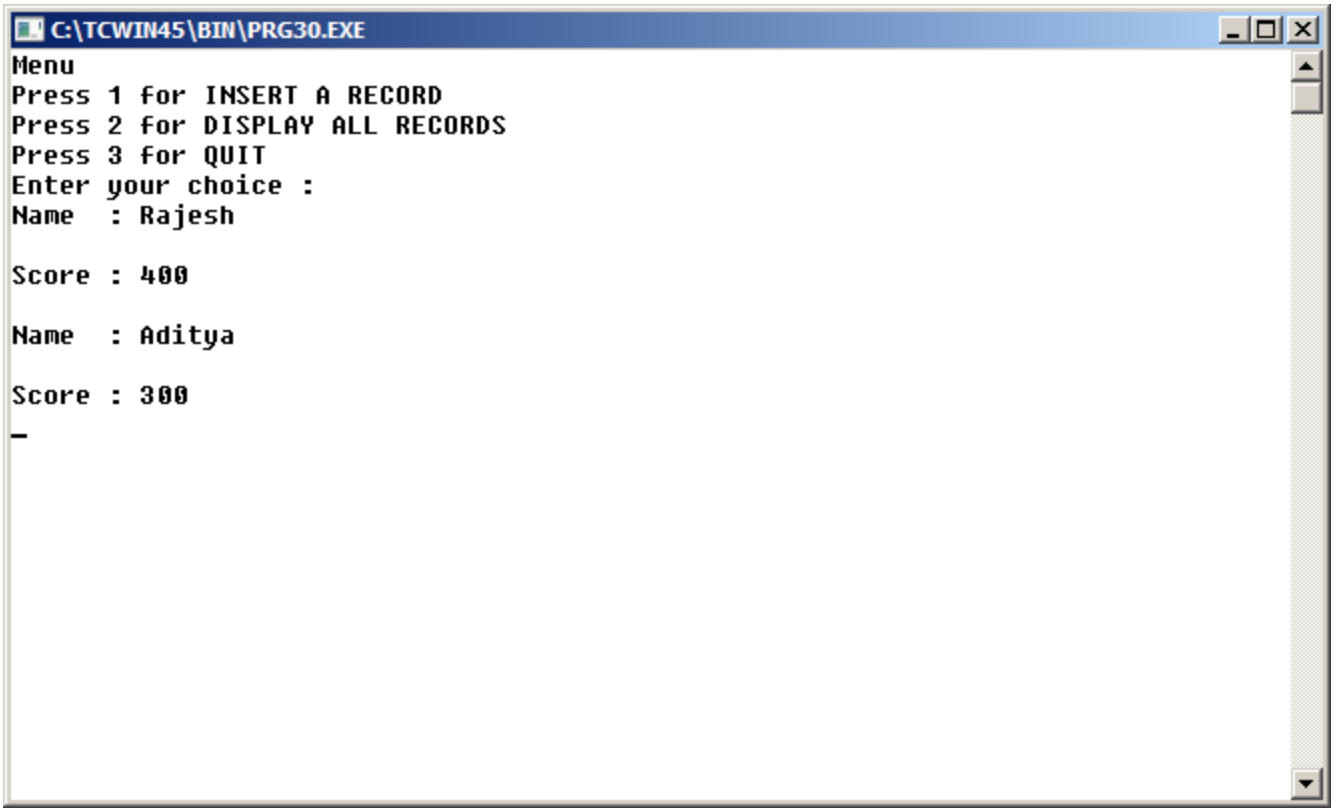
```
//    Program No. 30



#include<fstream.H>
#include<string.H>
#include<stdio.H>
#include<conio.H>


class Player
{
      char Name[50];
      int  Score;

        public:
      void Getdata();
      void Showdata();
      int  ReturnScore()
      { return Score;}
};


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



//    ********************** DISPLAY AN OBJECT ***************************
void Player::Showdata()
{
      cout<<"\nName   : "<<Name<<endl;
      cout<<"\nScore  : "<<Score<<endl;
}
```

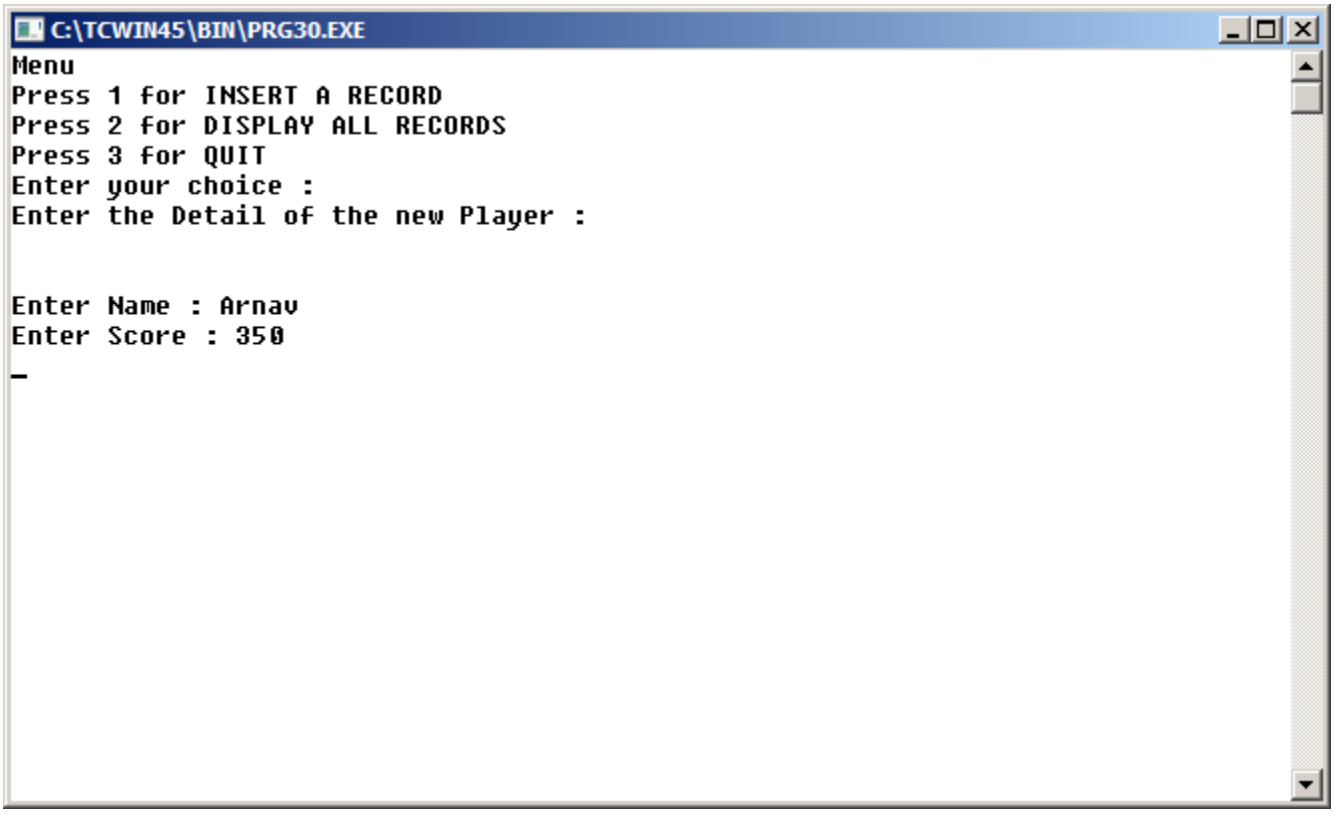
```
//   ******************* APPEND (ADD) A NEW RECORD **********************
void Insert()
{

     Player P, FP ;
     int Flag=0;

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

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

     if(f1)
     {
          while( f1.read((char*) &FP, sizeof(FP)) )
          {
            if(P.ReturnScore()>FP.ReturnScore() && Flag==0)
            {
                    f2.write((char *) &P, sizeof(P));
                    Flag=1;
            }
            f2.write((char *) &FP, sizeof(FP));
          }
     }
     else
     {
          f2.write((char *) &P, sizeof(P));
          f2.close();
     }
     f1.close();
     f2.close();

     remove("PLAYER.DAT");
     rename("TEMP.DAT" , "PLAYER.DAT");
}
```

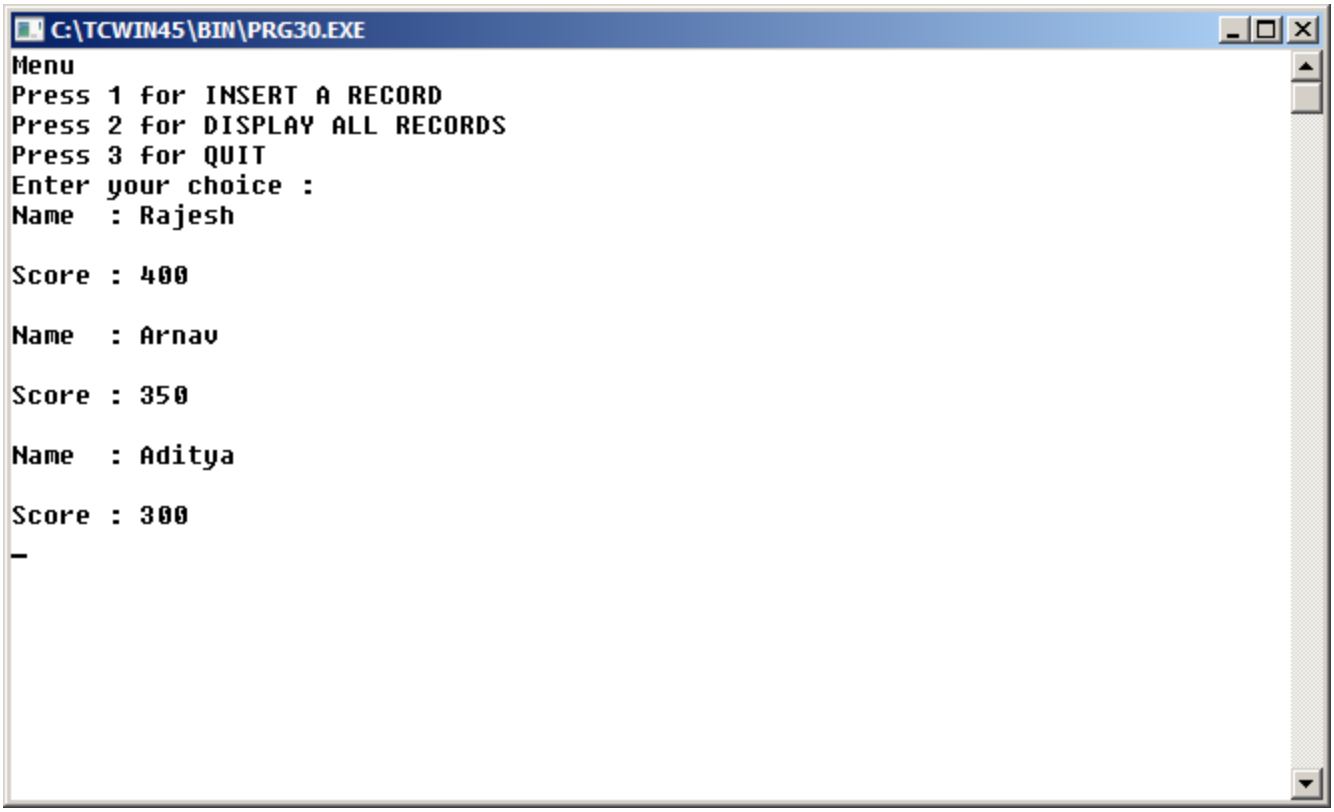
```
//   ********************* DISPLAY ALL RECORDS ***************************
void DisplayAll()
{

     Player P;

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

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

     f1.close();
}

void main()
{
     char choice;
     do{
          clrscr();
          cout<<"Menu\n";
          cout<<"Press 1 for INSERT A RECORD \n";
          cout<<"Press 2 for DISPLAY ALL RECORDS \n";
          cout<<"Press 3 for QUIT \n";
          cout<<"Enter your choice : ";
          choice=getch();
          switch(choice)
          {
               case '1': Insert(); break;
               case '2': DisplayAll();  break;
;
               case '3':           break;
               default :cout<<"\nWrong Choice Entered !!\n\n";
          }
          getch();
     }while(choice!='3');
}
```


C:\TCWIN45\BIN\PRG30.EXE
Menu
Press 1 for INSERT A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for QUIT
Enter your choice :
Enter the Detail of the new Player :
Enter Name : Aditya
Enter Score : 300


C:\TCWIN45\BIN\PRG30.EXE
Menu
Press 1 for INSERT A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for QUIT
Enter your choice :
Enter the Detail of the new Player :
Enter Name : Rajesh
Enter Score : 400


C:\TCWIN45\BIN\PRG30.EXE
Menu
Press 1 for INSERT A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for QUIT
Enter your choice :
Name : Rajesh
Score : 400
Name : Aditya
Score : 300


C:\TCWIN45\BIN\PRG30.EXE
Menu
Press 1 for INSERT A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for QUIT
Enter your choice :
Enter the Detail of the new Player :
Enter Name : Arnav
Enter Score : 350


C:\TCWIN45\BIN\PRG30.EXE
Menu
Press 1 for INSERT A RECORD
Press 2 for DISPLAY ALL RECORDS
Press 3 for QUIT
Enter your choice :
Name : Rajesh
Score : 400
Name : Arnav
Score : 350
Name : Aditya
Score : 300

SQL TABLES

SQL : 1

Table: BOOKS

BOOK_ID	BOOK_NAME	AUTHOR_NAME	PUBLISHERS	PRICE	TYPE	QUANTITY
C0001	Fast Cook	Lata Kapoor	EPB	355	Cookery	5
F0001	The Tears	William Hopkins	First Publishers	650	Fiction	20
T0001	My First C++	Brain & Brroke	EPB	350	Text	10
T0002	C++ Brainworks	A.M.Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First Publishers	750	Fiction	50

Table: ISSUED

BOOK_ID	QUANTITY_ISSUED
T0001	4
C0001	5
F0001	2

```
mysql> select * from BOOks;
+---------+----------------+-----------------+------------------+-------+-------
--+----------+
| BOOK_ID | BOOK_NAME      | AUTHOR_NAME     | PUBLISHERS       | PRICE | TYPE
  | QUANTITY |
+---------+----------------+-----------------+------------------+-------+-------
--+----------+
| C0001   | Fast Cook      | Lata Kapoor     | EPB              |   355 | Cooker
y |        5 |
| F0001   | The Tears      | William Hopkins | First Publishers |   650 | Fictio
n |       20 |
| T0001   | My First C++   | Brain & Brroke  | EPB              |   350 | Text
  |       10 |
| T0002   | C++ Brainworks | A.M. Rossaine   | TDH              |   350 | Text
  |       15 |
| T0002   | C++ Brainworks | A.M. Rossaine   | TDH              |   350 | Text
  |       15 |
| F0002   | Thunderbolts   | Anna Roberts    | First Publishers |   750 | Fictio
n |       50 |
+---------+----------------+-----------------+------------------+-------+-------
--+----------+
```

```
mysql> select * from ISSUED;
+---------+-----------------+
| BOOK_ID | QUANTITY_ISSUED |
+---------+-----------------+
| T0001   |               4 |
| C0001   |               5 |
| F0001   |               2 |
+---------+-----------------+
3 rows in set (0.00 sec)
```

Queries based on Tables BOOKS & ISSUED

1. To show book name, author name and price of books of First Publishers.

   Select BOOK_NAME, AUTHOR_NAME, PRICE From BOOKS Where PUBLISHERS=”First Publishers”;

2. To list the names from books of Text type.

   Select BOOK_NAME From BOOKS Where TYPE=”Text”;

3. To display the names and price from books in ascending order of their price.

   Select BOOK_NAME, PRICE From BOOKS Order By PRICE;

4. To increase the price of all books of EPB publishers by 10%.

   Update BOOKS Set PRICE=PRICE*1.1Where PUBLISHERS=”EPB”;

5. To display the book id, book name and quantity issued for all books which have been issued.

   Select BOOK_ID, BOOK_NAME, QUANTITY_ISSUED From BOOKS, ISSUED

   Where ISSUED.BOOK_ID=BOOKS.BOOK_ID;

6. To insert new row in the table issued having the following data:

   ‘F3003’,1

   Insert Into ISSUED

   Fields (BOOK_ID, QUANTITY_ISSUED)

   Values(“F3003”,1);

7. To display the total no. of books (quantity) of each type.

   Select TYPE, Sum(QUANTITY) From BOOKS Group By TYPE;

8. To find the maximum price from books which have quantity more than 15.

   Select Max(PRICE) From BOOKS Where QUANTITY>15;

9. To count unique publishers from books & cost less than 400.

   Select Count(Distinct PUBLISHERS) From BOOKS Where PRICE<400;

10. To display the no. of titles available of each publishers.

    Select PUBLISHERS, Count(*) From BOOKS Group By PUBLISHERS;

SQL : 2

TABLE :ITEM

I_ID	ItemName	Manufacturer	Price
PC01	Personal Computer	ABC	35000
LC05	Laptop	ABC	55000
PC03	Personal Computer	XYZ	32000
PC06	Personal Computer	COMP	37000
LC03	Laptop	PQR	57000

TABLE:CUSTOMER

C_ID	CustomerName	City	I_Id
01	N Roy	Delhi	LC03
06	H Singh	Mumbai	PC03
12	R Pandey	Delhi	PC06
15	C Sharma	Delhi	LC03
16	K Agarwal	Bangalore	PC01

```
mysql> select * from ITEM;
+------+--------------------+--------------+-------+
| I_ID | ItemName           | Manufacturer | Price |
+------+--------------------+--------------+-------+
| PC01 | Personal Computer  | ABC          | 35000 |
| LC01 | Laptop             | ABC          | 55000 |
| PC03 | Personal Computer  | XYZ          | 32000 |
| PC06 | Personal Computer  | COMP         | 37000 |
| LC03 | Laptop             | PQR          | 57000 |
+------+--------------------+--------------+-------+
5 rows in set (0.00 sec)
```

```
mysql> select * from customer;
+------+--------------+-----------+------+
| C_ID | CustomerName | City      | I_ID |
+------+--------------+-----------+------+
| 01   | N ROY        | Delhi     | LC03 |
| 06   | H Singh      | Mumbai    | PC03 |
| 12   | R Pandey     | Delhi     | PC06 |
| 15   | C Sharma     | Delhi     | LC03 |
| 16   | K Agarwal    | Bangalore | PC01 |
+------+--------------+-----------+------+
5 rows in set (0.00 sec)
```

Queries based on Tables ITEM & CUSTOMER

1. To display the details of those Customers whose City is Delhi.

   SELECT * FROM CUSTOMER WHERE City='Delhi';

2. To display the details of Items whose Price is in the range of 35000 to 55000 (Both values included)

   SELECT * FROM ITEM WHERE PRICE BETWEEN 35000 AND 55000;

3. To display the CustomerName, City from table Customer and ItemName and Price from table Item with their corresponding matching I_Id.

   SELECT CustomerName,City,ItemName,Price

   FROM CUSTOMER,ITEM WHERE CUSTOMER.I_Id=ITEM.I_ID;

4. To increase the Price of all Items by 1000 in the table Item.

   UPDATE ITEM SET PRICE=PRICE+1000;

5. To display city wise number of customers.

   SELECT City, COUNT(CustomerName) FROM CUSTOMER GROUP BY City;

6. To display the number of products manufactured by each Manufacturers

   SELECT Manufacturer, COUNT(ItemName) FROM ITEM GROUP BY Manufacturer;

7. To display the ItemName and the CustomerName who brought that item.

   SELECT CustomerName, ItemName

   FROM CUSTOMER, ITEM WHERE CUSTOMER.I_Id=ITEM.I_ID;

8. To reduce the price of Items by 10% manufactured by the company 'ABC'.

   UPDATE ITEM SET PRICE=PRICE*0.9 WHERE manufacturer='ABC';

9. To display all the items in the order of their Price (Low to High).

   SELECT * FROM ITEM ORDER BY PRICE;

10. Add a new attribute YEAR in the table ITEM to store the year of manufacturing.

    ALTER TABLE ITEM
    ADD (YEAR Integer);