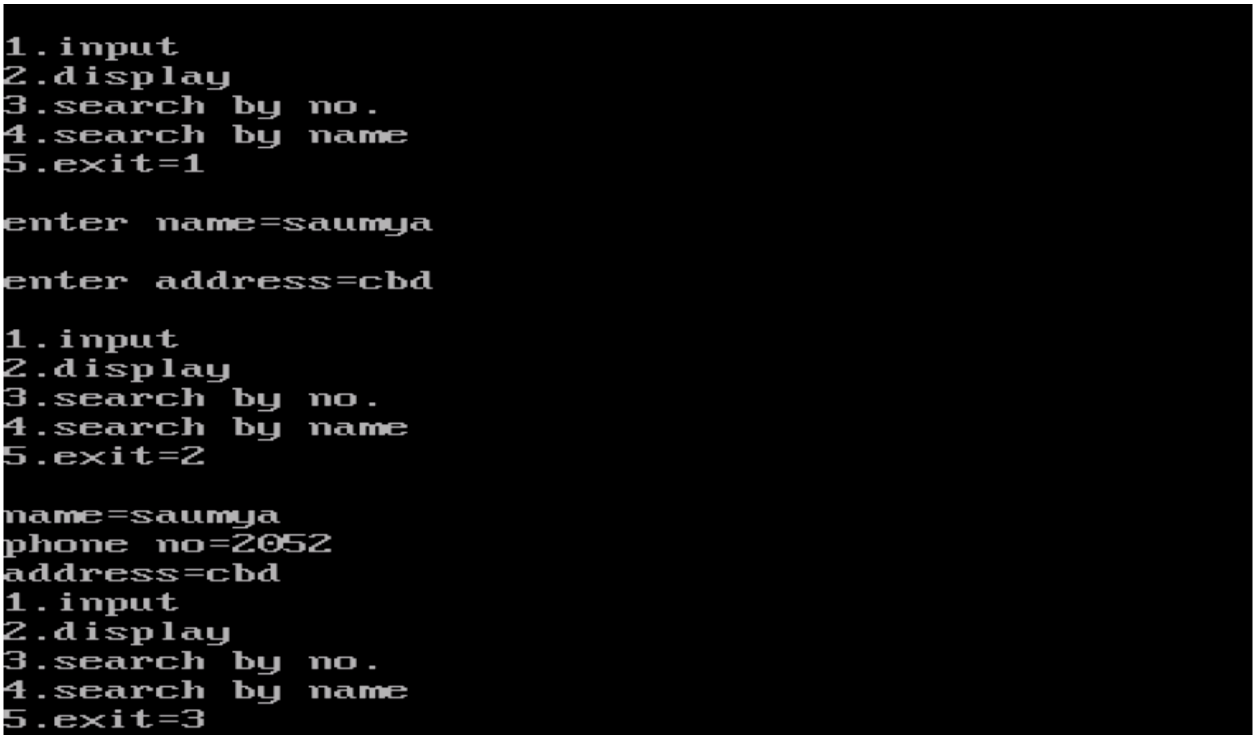
1 CLASS DESIGN-Telephone directory (IMPLEMENT ARRAY OF OBJECTS)

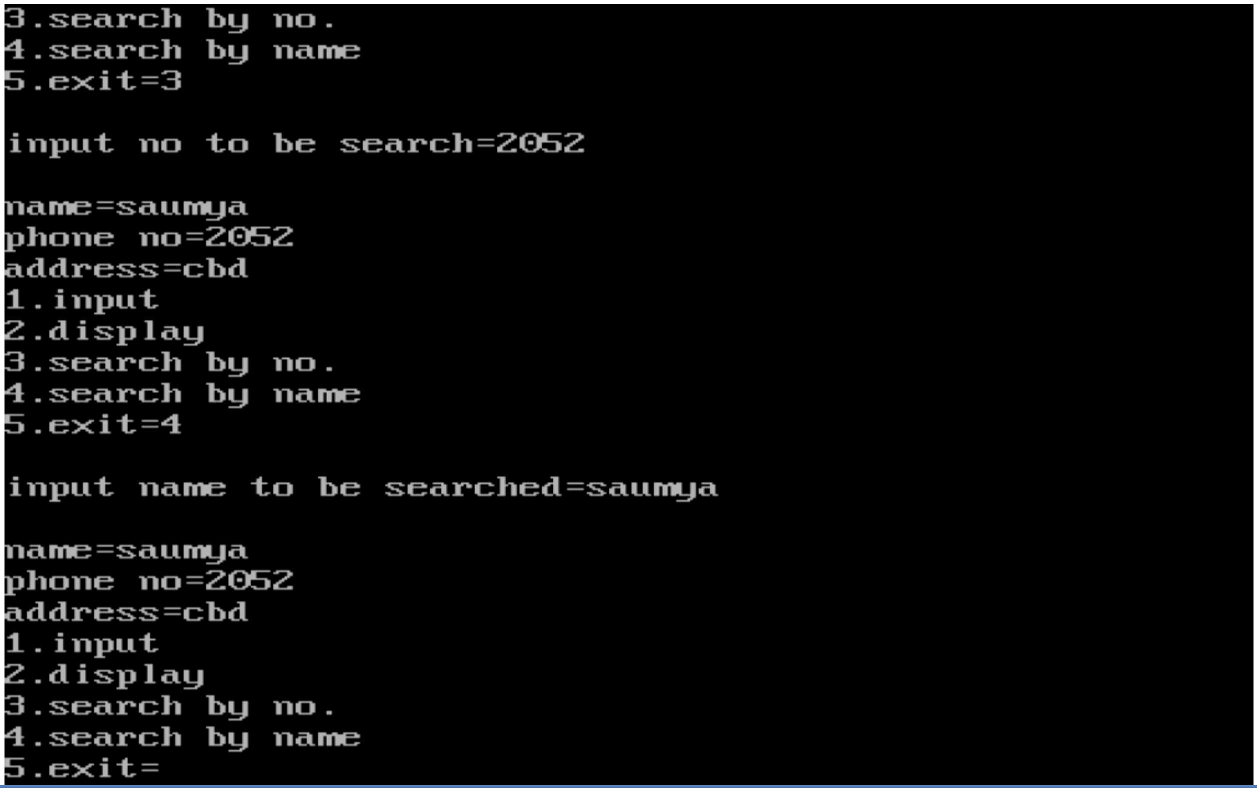
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
class directory
{
 long int phoneno;
 char cusname[30];
 char cusadd[100];
 public:
 void input();
 void display();
 long int getno()
  {
    return phoneno;
  }
 char*getname()
  {
    return cusname;
  }
};
void directory::input()
{
 cout<<"\nenter name=";
 gets(cusname);
 cout<<"\nenter address=";
 gets(cusadd);
 phoneno=random(7000-2000+1)+2000;
}
void directory::display()
{
  cout<<"\nname="<<cusname<<"\nphone no="<<phoneno<<"\naddress="<<cusadd;
}
 void main()
{
  clrscr();
  int menu,cnt=0,x;
```

```
      directory d[50];

     do
      {
        cout<<"\n1.input\n2.display\n3.search by no.\n4.search by name\n5.exit=";
        cin>>menu;
        switch(menu)
       {
         case 1: d[cnt].input();cnt++;break;
          case 2:for(x=0;x<cnt;x++)
                     d[x].display();
                      break;
          case 3:long no;
                      int flag=0;
                      cout<<"\ninput no to be search=";
                      cin>>no;
                      for(x=0;x<cnt;x++)
                      {
                          if(no==d[x].getno())
                          {
                             d[x].display();flag=1;break;
                           }
                       }
                       if (flag==0)
                       cout<<"\nnot found";
                       break;
           case 4:char str[30];
                      cout<<"\ninput name to be searched=";
                      gets(str);
                      flag=0;
                      for(x=0;x<cnt;x++)
                      {
                        if(!strcmpi(str,d[x].getname()))
                         {
                           d[x].display();
                           flag=1;
                           break;
                         }
                     }
                      if(flag==0)
                     cout<<"\nnot found";
                    break;
```

```
            }
}while(menu<=4);
getch();
}
```



```
1.input
2.display
3.search by no.
4.search by name
5.exit=1

enter name=saumya

enter address=cbd

1.input
2.display
3.search by no.
4.search by name
5.exit=2

name=saumya
phone no=2052
address=cbd
1.input
2.display
3.search by no.
4.search by name
5.exit=3
```

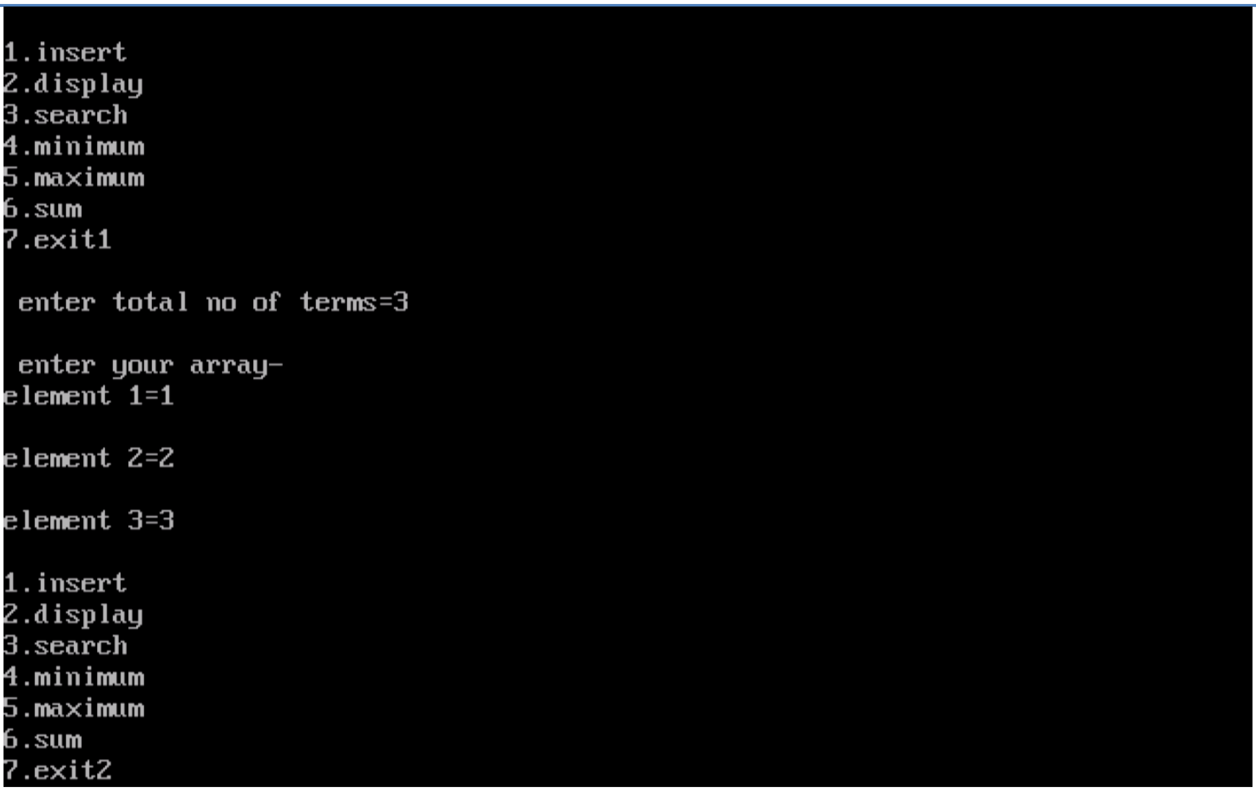


```
3.search by no.
4.search by name
5.exit=3

input no to be search=2052

name=saumya
phone no=2052
address=cbd
1.input
2.display
3.search by no.
4.search by name
5.exit=4

input name to be searched=saumya

name=saumya
phone no=2052
address=cbd
1.input
2.display
3.search by no.
4.search by name
5.exit=
```

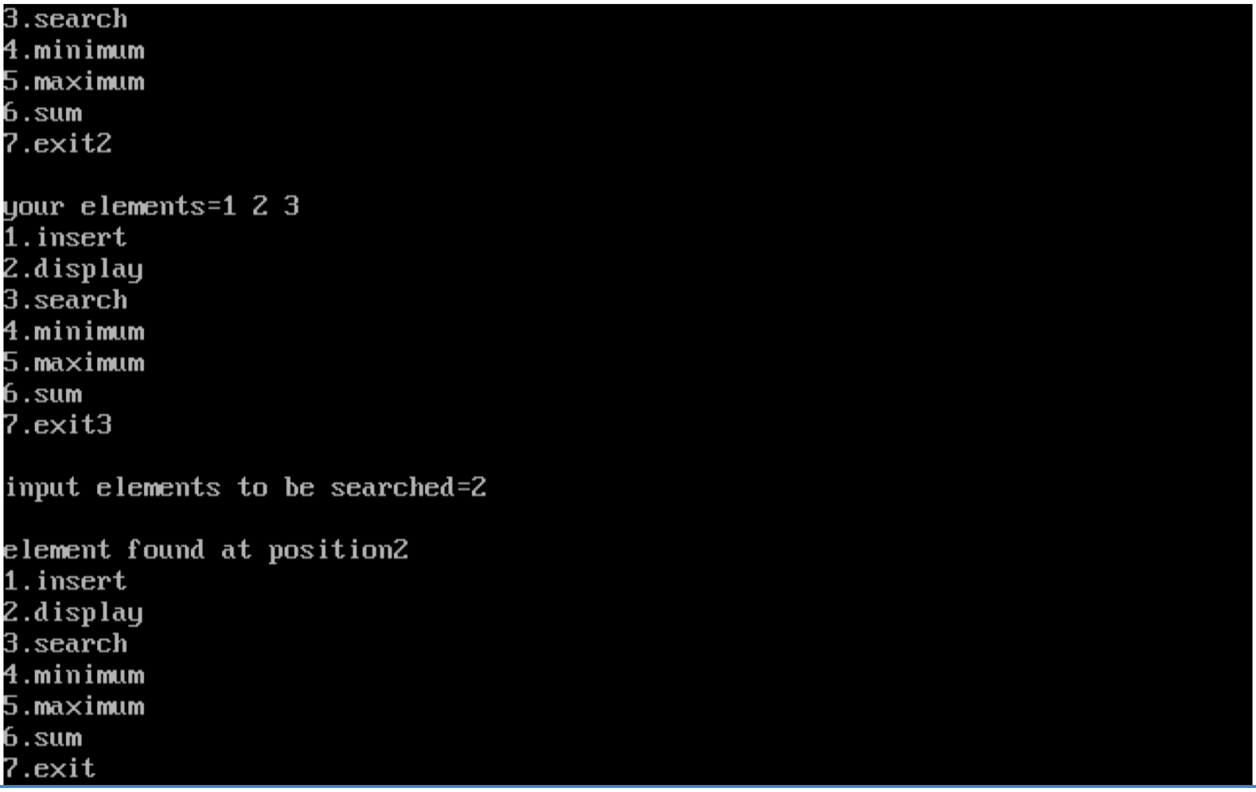
2. Create a class in c++ which maintains an integer array and provides following functions on it . Make a menu based options which includes:
1.Insert 2.Display 3.Search 4.Minimum 5.Maximum 6.Sum

```
#include<iostream.h>
#include<conio.h>
class array
{
 int arr[100],n;
 public:
array()
 {
  n=0;
 }
 void input();
 void display();
 void search(int n);
 int min();
 int max();
 int sum();
};
void array::input()
{
cout<<"\n enter total no of terms=";
cin>>n;
cout<<"\n enter your array-";
for(int a=0;a<n;a++)
 {
cout<<"\nelement "<<a+1<<"=";
cin>>arr[a];
 }
}
void array::display()
{
cout<<"\nyour elements=";
for(int a=0;a<n;a++)
 {
cout<<arr[a]<<" ";
 }
}
void array::search(int a)
{
 int flag=0;
for(int k=0;k<n;k++)
 {
  if(a==arr[k])
  {
cout<<"\nelement found at position"<<k+1;
    flag++;
  }
 }
 if(flag==0)
 {
cout<<"\nElement not found";
 }
}
int array::min()
{
 int p;
 p=arr[0];
for(int a=1;a<n;a++)
 {
  if(arr[a]<p)
  {
   p=arr[a];
  }
 }
 return p;
}
int array::max()
{
 int m=0;
for(int x=0;x<n;x++)
 {
  if(arr[x]>m)
  m=arr[x];
 }
 return m;
}
int array::sum()
{
 int s=0;
for(int x=0;x<n;x++)
 {
  s+=arr[x];
 }
 return s;
}
```

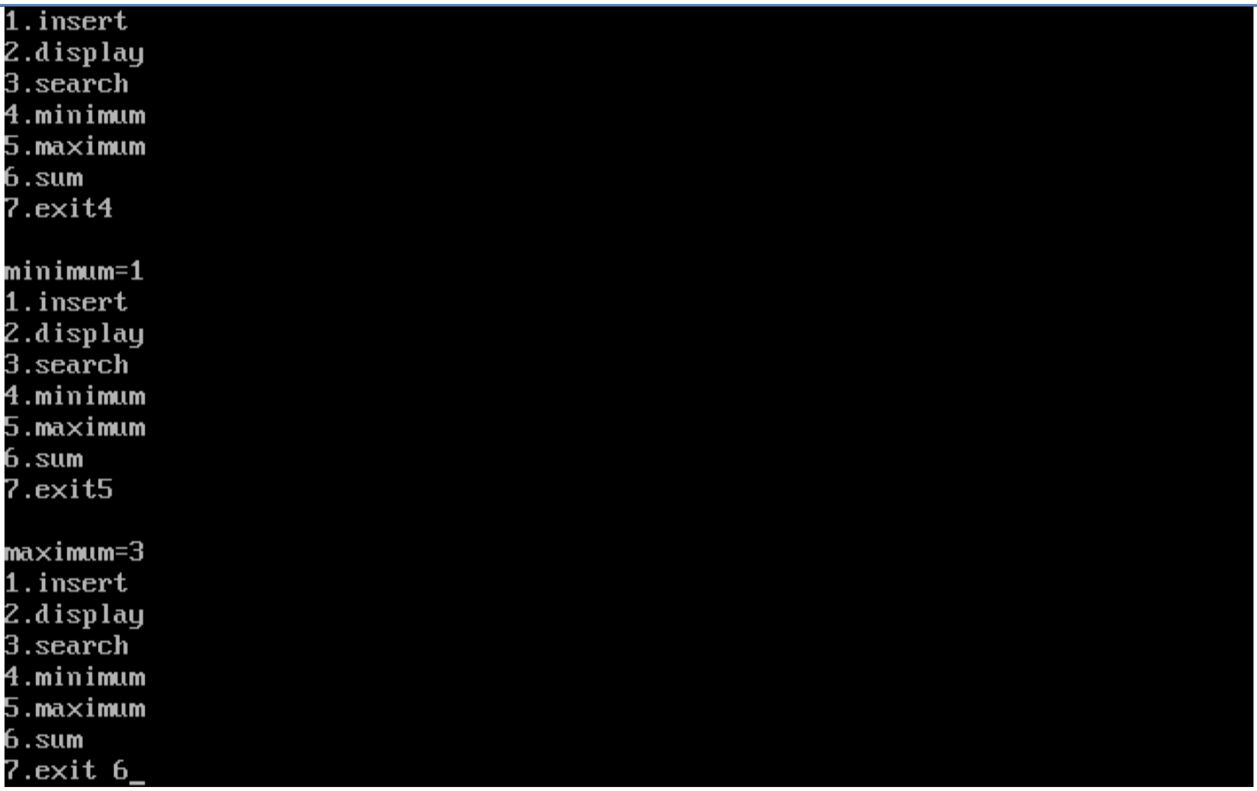
```
void main()
{
clrscr();
 array a;
 int menu,x;
 do
 {
cout<<"\n1.insert \n2.display \n3.search  \n4.minimum  \n5.maximum \n6.sum \n7.exit";
cin>>menu;
  switch(menu)
  {
   case 1:a.input();
            break;
   case 2:a.display();
            break;
   case 3:cout<<"\ninput elements to be searched=";
          cin>>x;
          a.search(x);
            break;
   case 4:cout<<"\nminimum="<<a.min();
            break;
   case 5:cout<<"\nmaximum="<<a.max();
            break;
   case 6:cout<<"\ntotal sum="<<a.sum();
            break;
  }
}while(menu!=7);
getch();
}
```



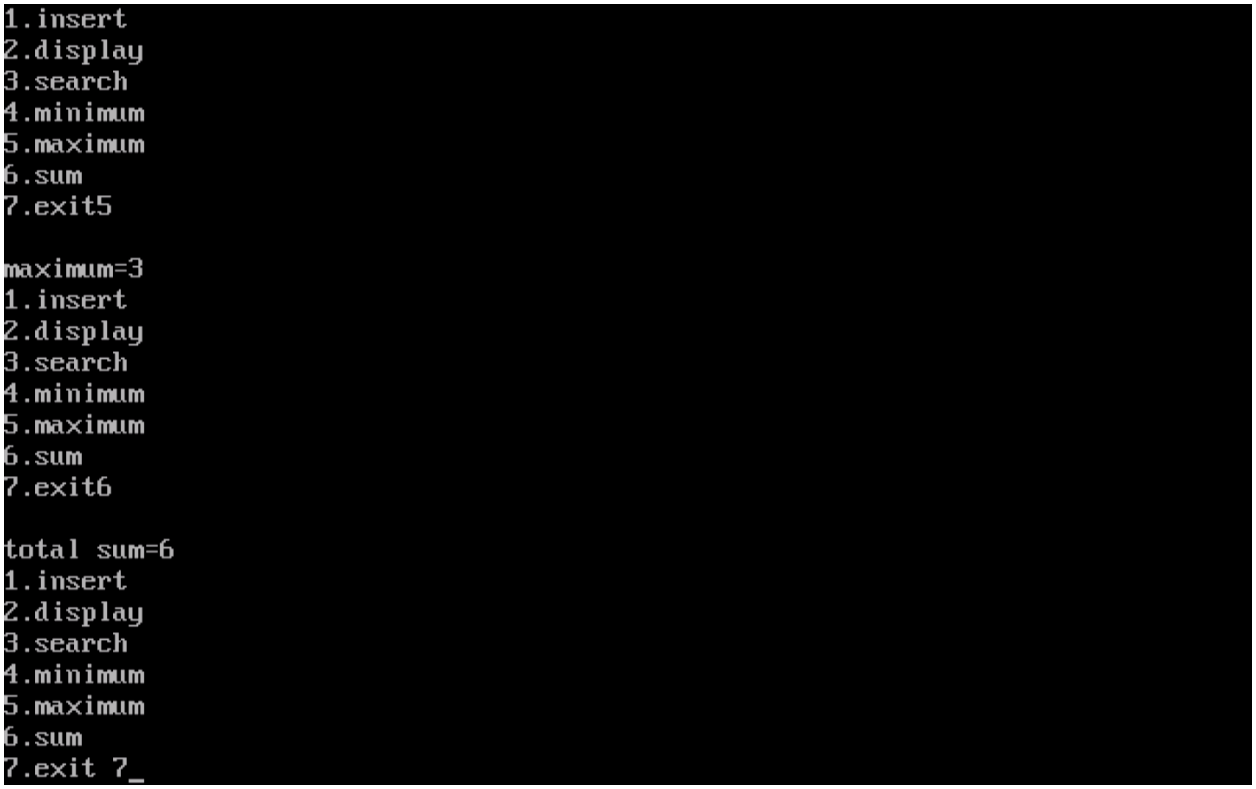
```
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit1

 enter total no of terms=3

 enter your array-
element 1=1

element 2=2

element 3=3

1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit2
```



```
3.search
4.minimum
5.maximum
6.sum
7.exit2

your elements=1 2 3
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit3

input elements to be searched=2

element found at position2
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit
```



```
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit4

minimum=1
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit5

maximum=3
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit 6_
```



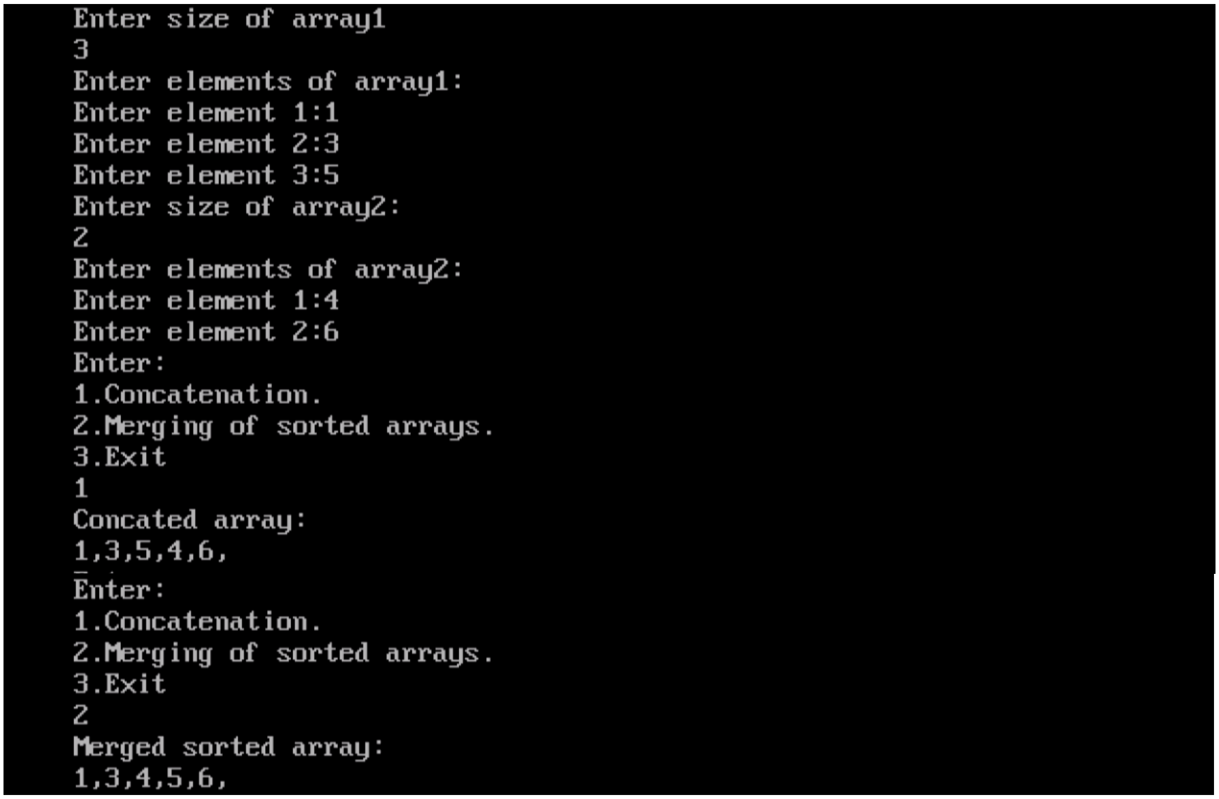
```
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit5

maximum=3
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit6

total sum=6
1.insert
2.display
3.search
4.minimum
5.maximum
6.sum
7.exit 7_
```

3. Program to implement following operations on 1D array
1.Concatenation of two arrays
2.Merging of two sorted arrays

```
#include<iostream.h>
#include<conio.h>
class array
{
private:
          int c[20];
          int z,l;
          int d[30],a;
          int x,y;
public:
          void concat(int arr[20],int m,int brr[20],int n);
          void merge(int arr[10],int m,int brr[10],int n);
};
void array::concat(int arr[20],int m,int brr[20],int n)
{
   z=0;l=m+n;
  for(int x=0;x<m;x++)
  {
    c[z]=arr[x];
    z++;
  }
for(int y=0;y<n;y++)
 {
   c[z]=brr[y];
   z++;
 }
 cout<<"Concated array:"<<endl;
 for(int s=0;s<l;s++)
 {
   cout<<c[s]<<",";
 }
 cout<<endl;
}
void array::merge(int arr[10],int m,int brr[10],int n)
{
 x=0;y=0;a=0;
while(x<m && y<n)
 {
```

```
 if(arr[x]<brr[y])
  {
   d[a]=arr[x];
   x++; a++;
  }
else
  {
   d[a]=brr[y];
    y++; a++;
  }
 }
while(x<m)
 {
   d[a]=arr[x];
   x++; a++;
 }
while(y<n)
 {
   d[a]=brr[y];
   a++; y++;
 }
 cout<<"Merged sorted array:"<<endl;
 for(int s=0;s<(m+n);s++)
 {
   cout<<d[s]<<",";
 }
cout<<endl;
}
void main()
{
clrscr();
array obj;
int ans,x,y,arr[20],brr[20];
cout<<"Enter size of array1"<<endl;
cin>>x;
cout<<"Enter elements of array1:"<<endl;
for(int p=0;p<x;p++)
 {
   cout<<"Enter element "<<p+1<<":";
   cin>>arr[p];
 }
cout<<"Enter size of array2:"<<endl;
```

```
cin>>y;
cout<<"Enter elements of array2:"<<endl;
for(int q=0;q<y;q++)
 {
   cout<<"Enter element "<<q+1<<":";
   cin>>brr[q];
 }
do
 {
   cout<<"Enter:\n1.Concatenation.\n2.Merging of sorted arrays.\n3.Exit"<<endl;
   cin>>ans;
   switch(ans)
  {
    case 1:obj.concat(arr,x,brr,y);
              break;
    case 2:obj.merge(arr,x,brr,y);
              break;
  }
}while(ans!=3);
getch();
}
```



```
Enter size of array1
3
Enter elements of array1:
Enter element 1:1
Enter element 2:3
Enter element 3:5
Enter size of array2:
2
Enter elements of array2:
Enter element 1:4
Enter element 2:6
Enter:
1.Concatenation.
2.Merging of sorted arrays.
3.Exit
1
Concated array:
1,3,5,4,6,
Enter:
1.Concatenation.
2.Merging of sorted arrays.
3.Exit
2
Merged sorted array:
1,3,4,5,6,
```

4. Make a menu based program to perform operations on a 2D array matrix. Input size and matrix from the user, then perform the following operations. Write separate functions for each operation

1.Sum of rows

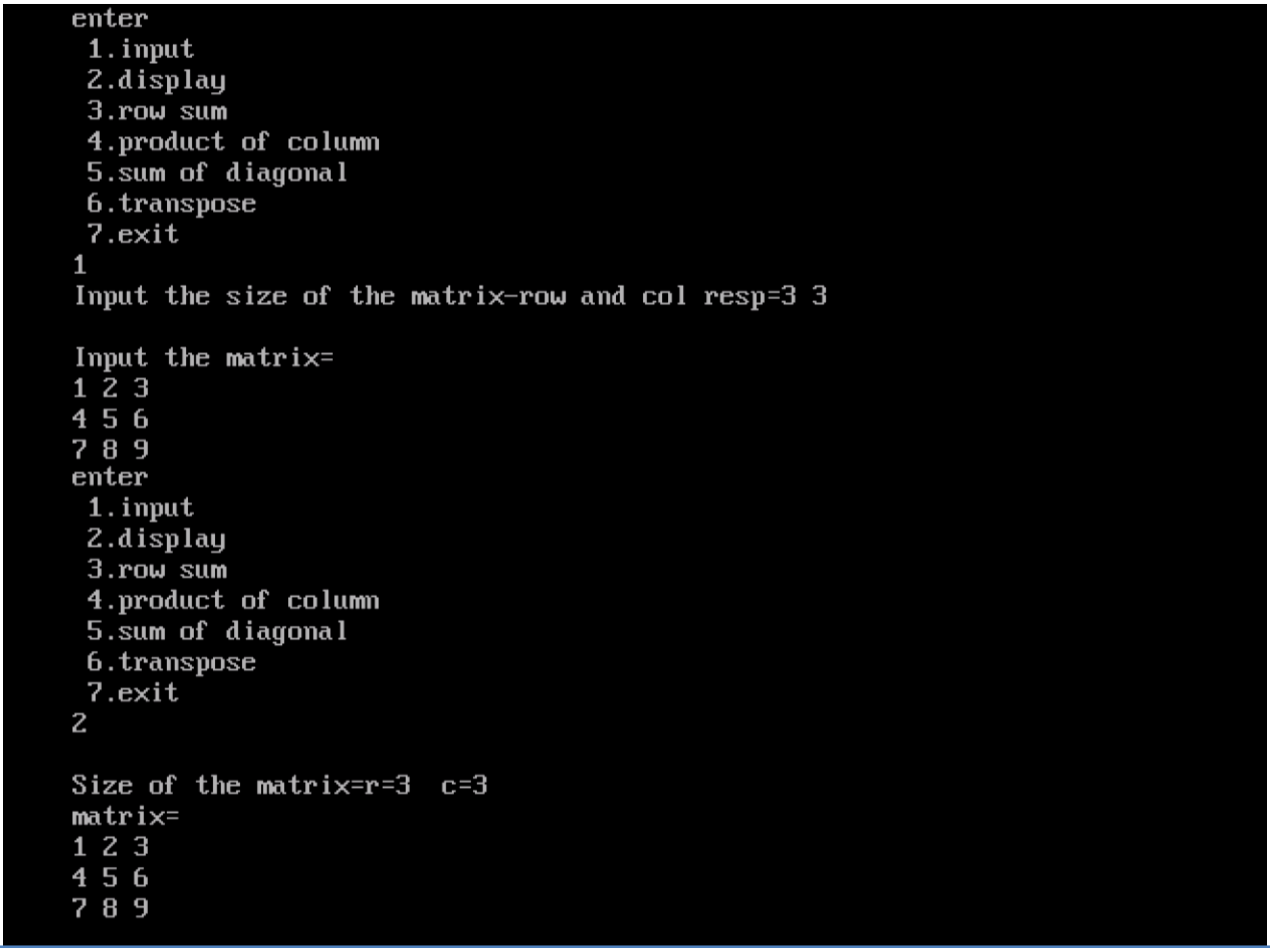
2.Product of Columns

3.Sum of diagonals

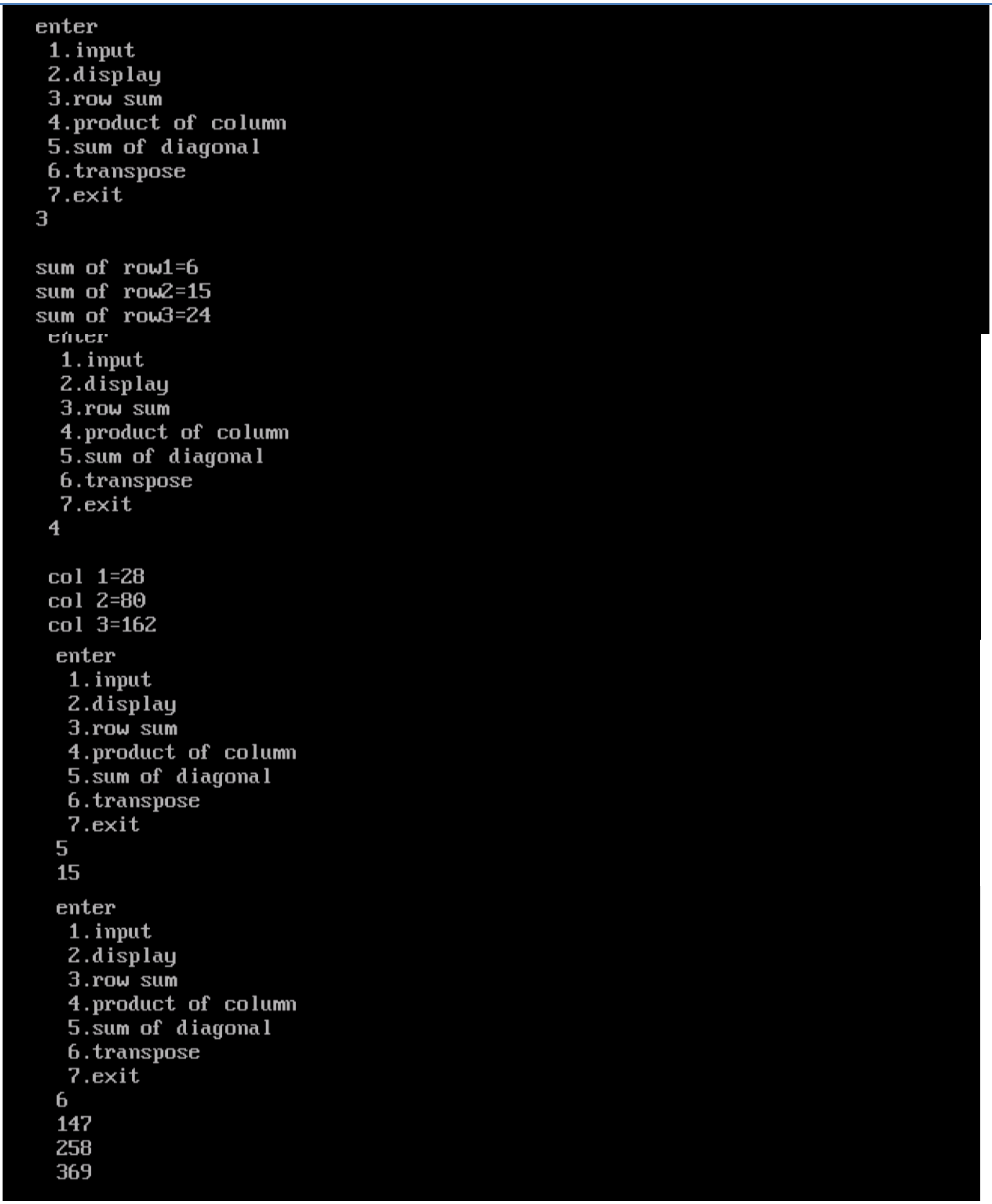
4.Transpose of matrix

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class matrix
{
private:
       int mat[5][5];
       int r,c;
public:
       void input();
       void display();
       void rowsum();
       void pcol();
       void diagonal();
       void transpose();
};
void matrix::input()
{
cout<<"Input the size of the matrix-row and col resp=";
cin>>r>>c;
cout<<"\nInput the matrix="<<endl;
for(int x=0;x<r;x++)
{
  for(int y=0;y<c;y++)
 {
   cin>>mat[x][y];
 }
}
}
void matrix::rowsum()
{
  int sum;
  for(int x=0;x<r;x++)
  {
    sum=0;
   for(int y=0;y<c;y++)
  {
    sum+=mat[x][y];
  }
  cout<<"\nsum of row"<<x+1<<"="<<sum;
 }
}
void matrix::pcol()
{
int product=1;
for(int y=0;y<r;y++)
{
  product=1;
  for(int x=0;x<c;x++)
  {
    product*=mat[x][y];
  }
  cout<<"\ncol "<<y+1<<"="<<product<<" ";
 }
}
void matrix::diagonal()
{
int dsum=0;
for(int x=0;x<r;x++)
{
 for(int y=0;y<c;y++)
  {
   if(x==y)
   dsum+=mat[x][y];
  }
 }
 cout<<dsum;
}
void matrix::display()
{
cout<<"\nSize of the matrix="<<"r="<<r<<"  c="<<c;
cout<<"\nmatrix="<<endl;
for(int x=0;x<r;x++)
{
  for(int y=0;y<c;y++)
  {
   cout<<mat[x][y]<<" ";
  }
  cout<<endl;
 }
}
void matrix::transpose()
{
int x,y,arr[5][5];
for(x=0;x<r;x++)
{
 for(y=0;y<c;y++)
 {
 arr[y][x]=mat[x][y];
 }
}
for(y=0;y<r;y++)
{
 for(x=0;x<c;x++)
 {
 cout<<arr[y][x];
 }
cout<<endl;
}
}

void main()
{
clrscr();
matrix m;
int menu;
char ch;
do
{
cout<<"\nenter\n 1.input \n 2.display \n 3.row sum \n 4.product of column \n 5.sum of
diagonal \n 6.transpose \n 7.exit"<<endl;
cin>>menu;
switch(menu)
{
case 1:m.input();
       break;
case 2:m.display();
       break;
case 3:m.rowsum();
```

```
        break;
case 4:m.pcol();
        break;
case 5:m.diagonal();
        break;
case 6:m.transpose();
        break;
}}while(menu!=7);

getch();
}
```



```
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
1
Input the size of the matrix-row and col resp=3 3

Input the matrix=
1 2 3
4 5 6
7 8 9
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
2

Size of the matrix=r=3  c=3
matrix=
1 2 3
4 5 6
7 8 9
```



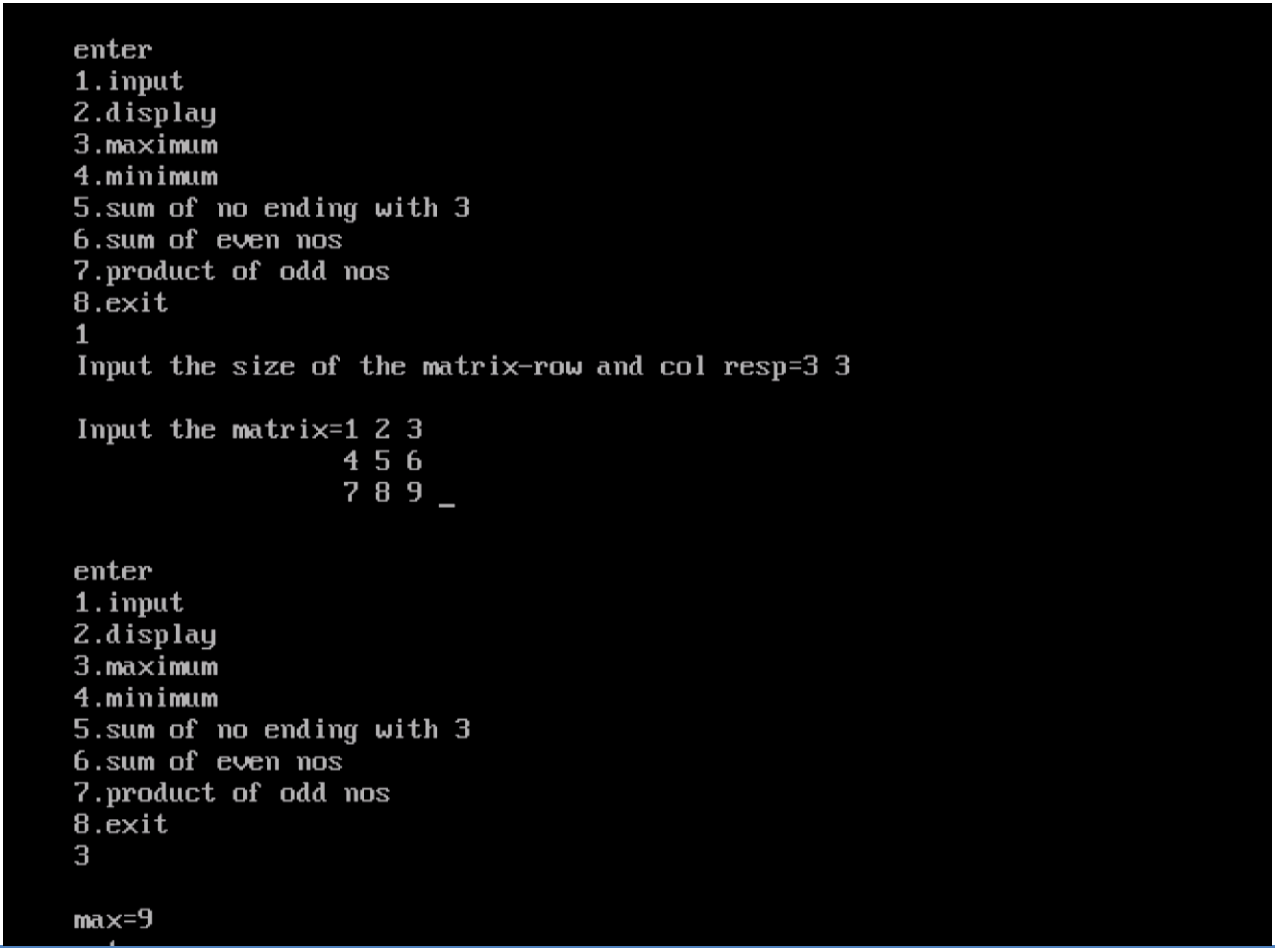
```
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
3

sum of row1=6
sum of row2=15
sum of row3=24
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
4

col 1=28
col 2=80
col 3=162
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
5
15
enter
 1.input
 2.display
 3.row sum
 4.product of column
 5.sum of diagonal
 6.transpose
 7.exit
6
147
258
369
```

5 Make a menu based program to perform operations on a 2D array matrix. Input size and matrix from the user, then perform the following operations

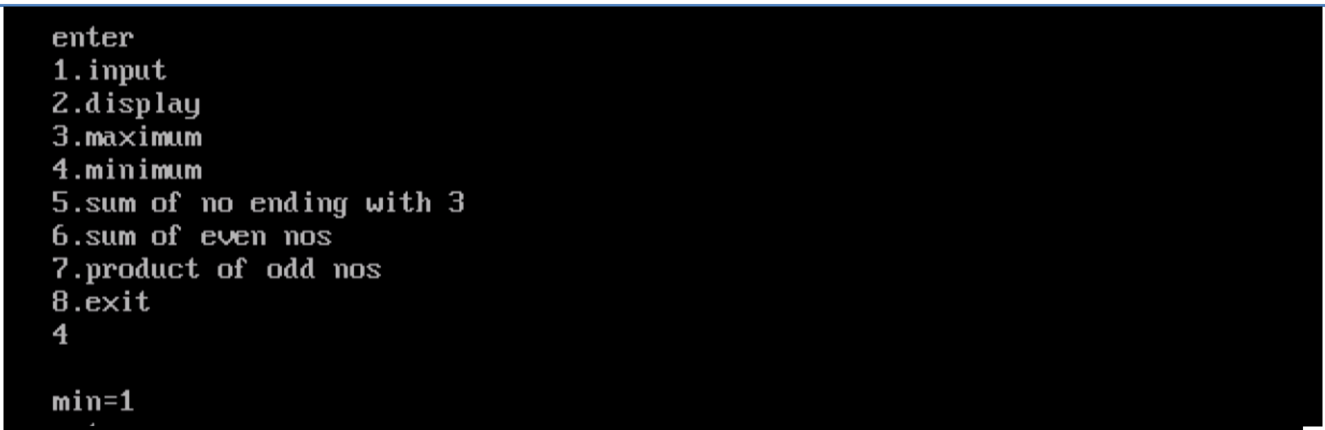
1. Maximum element in the array
2. Minimum element in the array
3. Find and display sum of elements ending with 3
4. Sum of Even numbers
5. Product of odd numbers in the array

```
#include<iostream.h>
#include<conio.h>
class matrix
{
int r,c, mat[5][5];
public:
       void input();
       void display();
       void maximum();
       void minimum();
       void sum3();
       void sumofeven();
       void podd();
};
void matrix::maximum()
{
 int x, y, max=mat[0][0];
 for (x=0;x<r;x++)
 {
  for(y=0;y<c;y++)
  {
   if(mat[x][y]>max)
   {
     max=mat[x][y] ;
   }
  }
}
cout<<"\nmax="<<max;
}
void matrix::minimum()
{
 int x,y,min=mat[0][0];
 for(x=0;x<r;x++)
 {
  for(y=0;y<c;y++)
 {
  if(mat[x][y]<min)
  min=mat[x][y];
  }
 }
 cout<<"\nmin="<<min;
}
void matrix::sum3()
{
 int x,y,sum=0;
 for(x=0;x<r;x++)
 {
  for(y=0;y<c;y++)
  {
   if((mat[x][y]%10)==3)
   sum+=mat[x][y];
  }
 }
 cout<<"\nsum="<<sum;
}
void matrix::sumofeven()
{
  int x,y,sum=0;
  for(x=0;x<r;x++)
 {
  for(y=0;y<c;y++)
 {
   if(mat[x][y]%2==0)
   sum+=mat[x][y];
  }
 }
 cout<<"sum of even="<<sum;
}
void matrix::podd()
{
 int x,y,p=1;
 for(x=0;x<r;x++)
 {
 for(y=0;y<c;y++)
 {
   if(mat[x][y]%2!=0)
   p*=mat[x][y];
 }
}
 cout<<"\nproduct of odd="<<p;
}
void matrix::input()
{
 cout<<"Input the size of the matrix-row and col resp=";
 cin>>r>>c;
 cout<<"\nInput the matrix=";
 for(int x=0;x<r;x++)
 {
  for(int y=0;y<c;y++)
 {
   cin>>mat[x][y];
 }
}
}
void matrix::display()
{
 cout<<"\nSize of the matrix="<<"r="<<r<<"  c="<<c;
 cout<<"\nmatrix="<<endl;
 for(int x=0;x<r;x++)
 {
  for(int y=0;y<c;y++)
  {
    cout<<mat[x][y]<<" ";
  }
  cout<<endl;
 }
}
void main()
{
  clrscr();
  matrix m;
  int menu;
  char ch;
  do
 {
  cout<<"\nenter\n1.input \n2.display \n3.maximum \n4.minimum \n5.sum of no ending
with 3 \n6.sum of even nos \n7.product of odd nos \n8.exit"<<endl;
 cin>>menu;
 switch(menu)
```

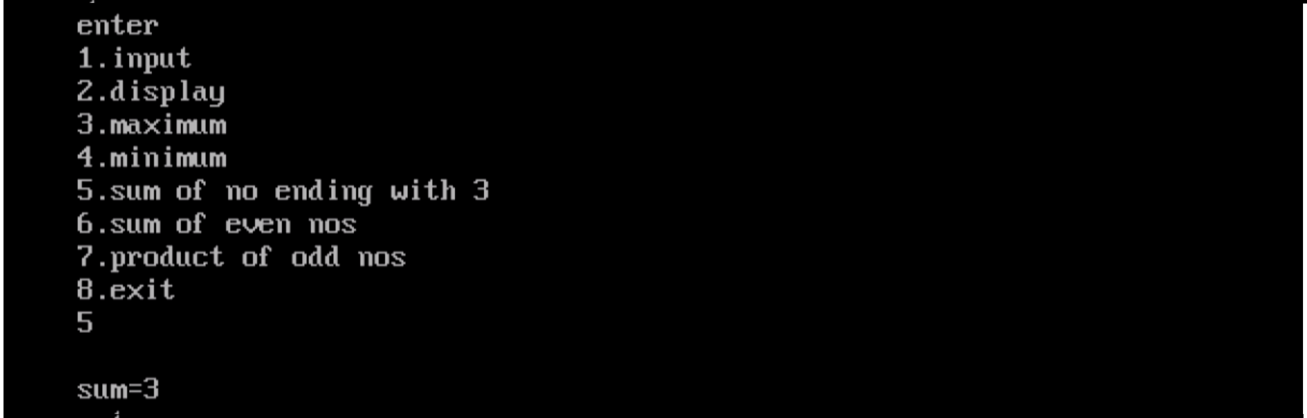
```
{
 case 1:m.input();
       break;
 case 2:m.display();
            break;
 case 3:m.maximum();
            break;
 case 4:m.minimum();
            break;
case 5:m.sum3();
            break;
case 6:m.sumofeven();
           break;
case 7:m.podd();
           break;
}}while(menu!=8);
getch();
}
```



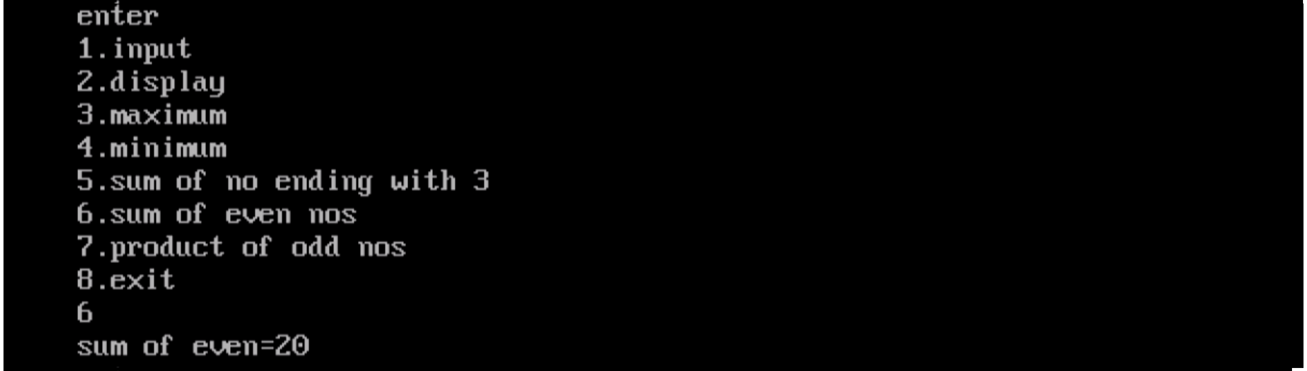
```
enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
1
Input the size of the matrix-row and col resp=3 3

Input the matrix=1 2 3
                 4 5 6
                 7 8 9 _


enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
3

max=9
```



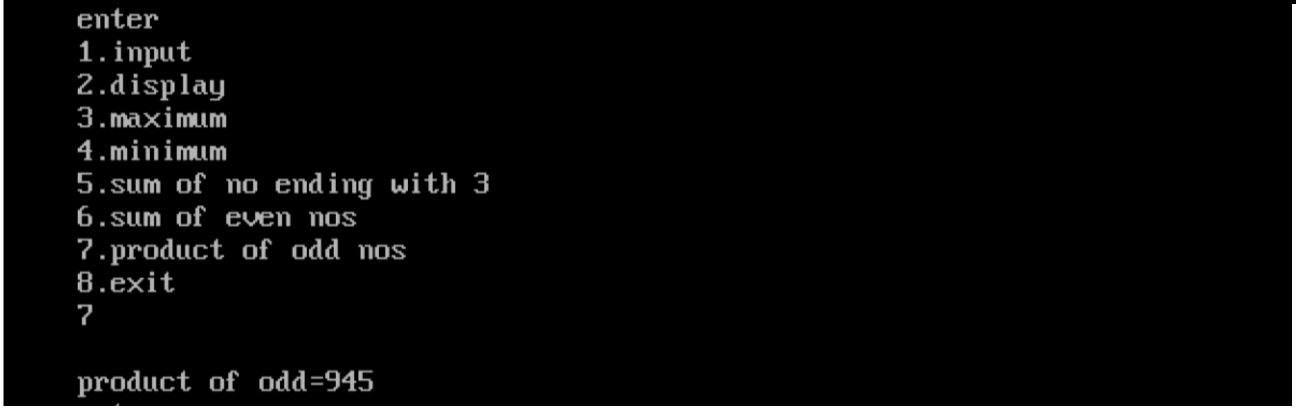
```
enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
4

min=1
```



```
enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
5

sum=3
```



```
enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
6
sum of even=20
```



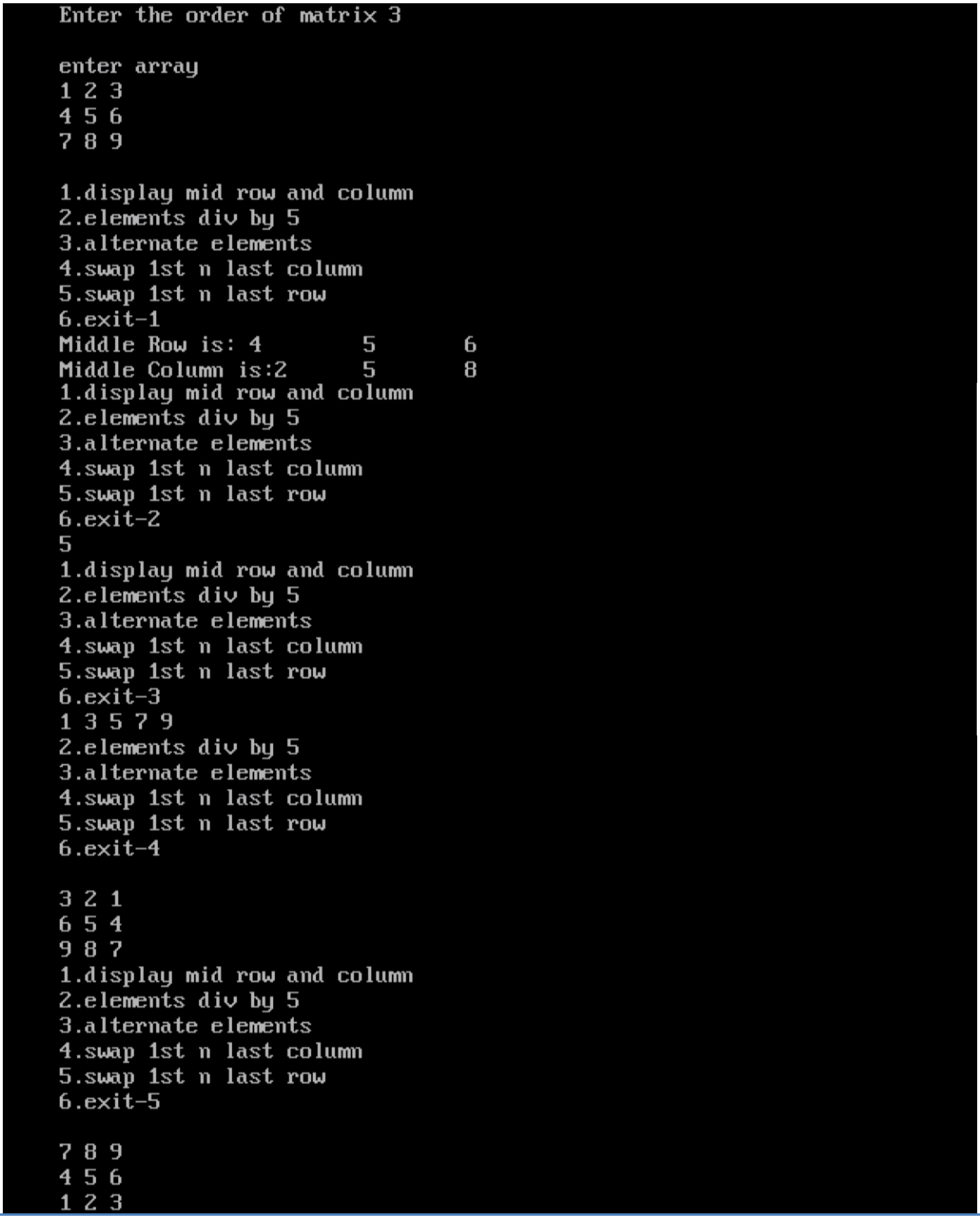
```
enter
1.input
2.display
3.maximum
4.minimum
5.sum of no ending with 3
6.sum of even nos
7.product of odd nos
8.exit
7

product of odd=945
```

6. **Make a menu based program to perform operations on a 2D array matrix. Input size and**
matrix from the user, then perform the following operations
1. Display elements of middle row and middle column
2. Display only those elements which are divisible by 5
3. Display all alternate elements in the array
4. Swap first column with the last column
5. Swap first row with the last row

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
void display(int arr[10][10],int size)
{
 for(int i=0;i<size;i++)
 {
  for (int j=0;j<size;j++)
  {
   cout<<arr[i][j]<<" ";
  }
  cout<<endl;
 }
}
void midrowcol(int arr[10][10],int size)
{
 int i,j;
 cout<<"Middle Row is: ";
  for (i=0;i<size;i++)
  {
   for( j=0;j<size;j++)
    {
    if (i== size/2)
    cout<<arr[i][j]<<"\t";
    }
  }
  cout<<"\nMiddle Column is:";
   for ( i=0;i<size;i++)
    {
     for( j=0;j<size;j++)
     {
     if(j==size/2)
     cout<<arr[i][j]<<"\t";
     }
    }
}
void div5(int arr[10][10],int size)
{
 for(int i=0;i<size;i++)
 {
  for(int j=0;j<size;j++)
  {
   if(arr[i][j]%5==0)
   { cout<<arr[i][j];}
  }
 }
}

void alternate(int arr[10][10],int n)
{
 for(int i=0;i<n;i++)
 {
  if(i%2==0)
   {
     for(int j=0;j<n;j+=2) cout<<arr[i][j]<<" ";
    }
   else
   {
     for(int j=1;j<n;j+=2) cout<<arr[i][j]<<" ";
   }
 }
}
void swapfirstlastrow(int m[10][10],int size)
{
          int rows=size;
          for (int i = 0; i <size; i++)
          {
              int t = m[0][i];
              m[0][i] = m[rows - 1][i];
              m[rows - 1][i] = t;
          }
          display(m,size);
}
void swapfirstlastcolumn(int m[10][10],int size)
```

```
{
int i,j,temp;
for (i=0; i<size; i++)
    {
     for(j=0; j<size; j++)
           {
             temp = m[i][1];
             m[i][1] = m[i][size];
             m[i][size] = temp;
           }
    }
    display(m,size);
}
int main()
{
clrscr();
 int size;
 cout<<"Enter the order of matrix ";
 cin>>size;
  int A[10][10];
  cout<<"\nenter array"<<endl;
  for(int i=0;i<size;i++)
  {
  for(int j=0;j<size;j++)
   {
    cin>>A[i][j];
   }
  }
  int menu;
 do
 {
 cout<<"\n1.display mid row and column\n2.elements div by 5\n3.alternate
elements\n4.swap 1st n last column \n5.swap 1st n last row\n6.exit-";
 cin>>menu;
 switch(menu)
 {
  case 1:midrowcol(A,size);break;
  case 2:div5(A,size);break;
  case 3:alternate(A,size);break;
  case 4:swapfirstlastcolumn(A,size);
          break;
  case 5:swapfirstlastrow(A,size);
```

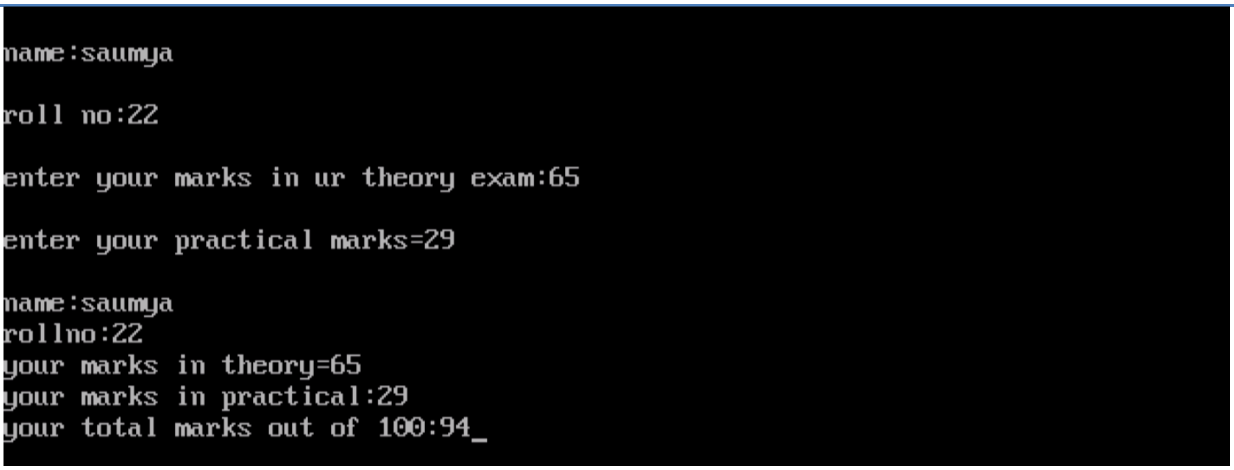
```
          break;
  case 6:exit(0);
 }
 }while(menu!=6);
getch();
 }
```



```
Enter the order of matrix 3

enter array
1 2 3
4 5 6
7 8 9

1.display mid row and column
2.elements div by 5
3.alternate elements
4.swap 1st n last column
5.swap 1st n last row
6.exit-1
Middle Row is: 4        5       6
Middle Column is:2      5       8
1.display mid row and column
2.elements div by 5
3.alternate elements
4.swap 1st n last column
5.swap 1st n last row
6.exit-2
5
1.display mid row and column
2.elements div by 5
3.alternate elements
4.swap 1st n last column
5.swap 1st n last row
6.exit-3
1 3 5 7 9
2.elements div by 5
3.alternate elements
4.swap 1st n last column
5.swap 1st n last row
6.exit-4

3 2 1
6 5 4
9 8 7
1.display mid row and column
2.elements div by 5
3.alternate elements
4.swap 1st n last column
5.swap 1st n last row
6.exit-5

7 8 9
4 5 6
1 2 3
```

7 MAKE A PROGRAM TO IMPLEMENT IS-ARELATIONSHIP (EXAM , PRACTICAL, THEORY, STUDENT)

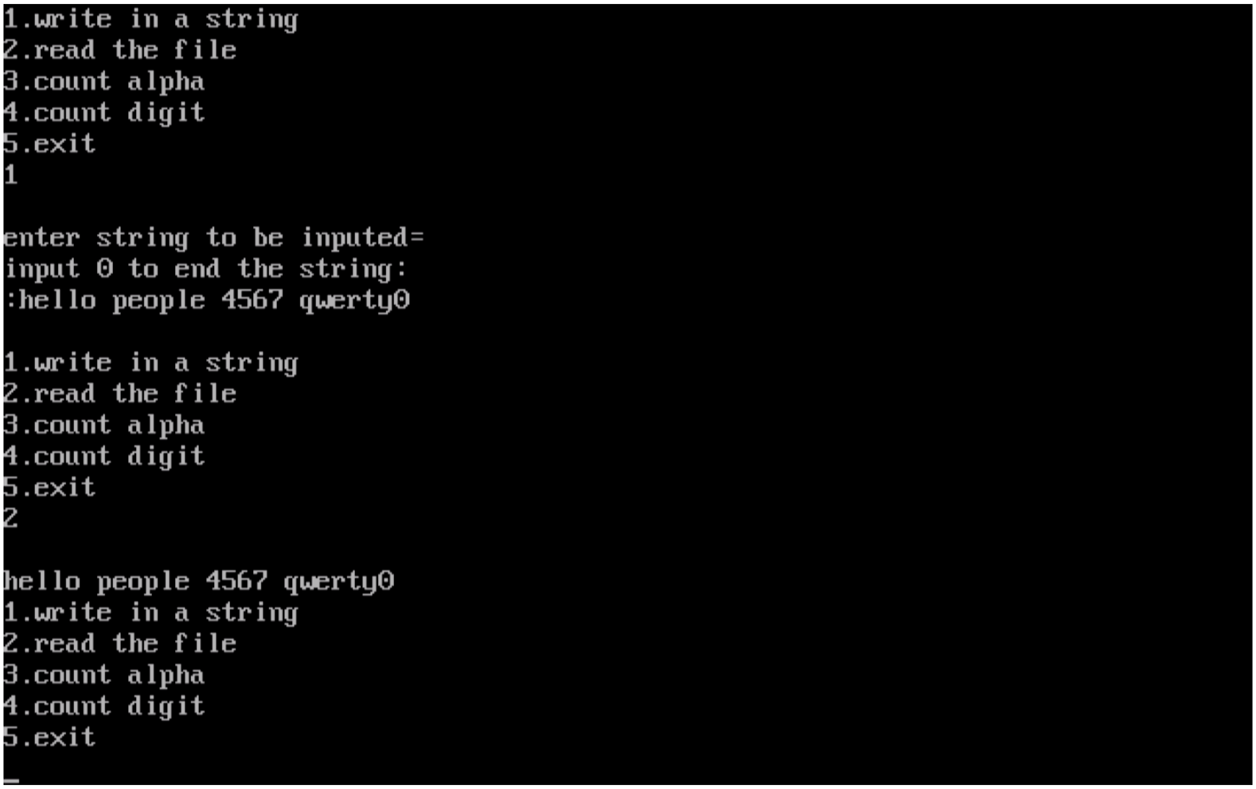
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class student
{
protected:int roll;
            char name[25];
public:voidgetdata()
       {
cout<<"\nname:";
cin>>name;
cout<<"\nroll no:";
cin>>roll;
       }
       void showdata()
       {
cout<<"\nname:"<<name<<"\nrollno:"<<roll;
       }
};
class theoryexam:virtual public student
{
protected:float marks;
public:voidreaddata()
       {
cout<<"\nenter your marks in ur theory exam:";
cin>>marks;
       }
       void showtheory()
       {
cout<<"\nyour marks in theory="<<marks;
       }
};
class practicalexam:virtual public student
{
protected:floatpmarks;
public:voidreadpr()
       {
cout<<"\nenter your practical marks=";
cin>>pmarks;
        }
        void showpr()
        {
cout<<"\nyour marks in practical:"<<pmarks;
        }
} ;
class result:publictheoryexam,publicpracticalexam
{
private:float total;
public:void calc()
        {
        total=marks+pmarks;
        }
        void display()
        {
cout<<"\nyour total marks out of 100:"<<total;
        }
};
void main ()
{
clrscr();
result r;
r.getdata();
r.readdata();
r.readpr();
r.calc();
r.showdata();
r.showtheory();
r.showpr();
r.display();
getch();
}
```



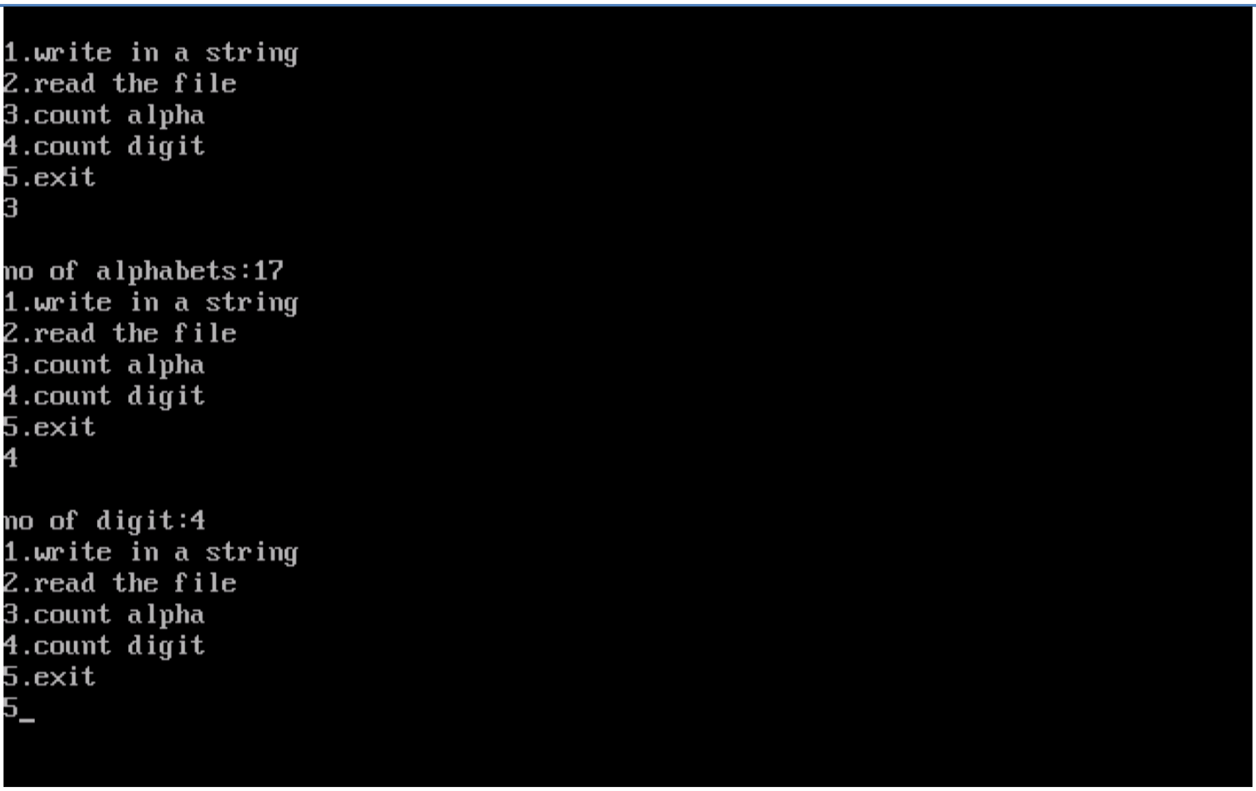
```
name:saumya

roll no:22

enter your marks in ur theory exam:65

enter your practical marks=29

name:saumya
rollno:22
your marks in theory=65
your marks in practical:29
your total marks out of 100:94_
```

8. **MAKE A MENU BASED PROGRAM TO PERFORM THE FOLLOWING OPERATIONS ON TEXT FILE (DATA.TXT) CHARCATER BY CHARACTER**
1-WRITE STRING IN THE FILE
2-READ
3-COUNT ALPHABETS
4-COUNT DIGITS

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<ctype.h>
void readdata()
{
 char ch;
 ifstream fin("data.txt",ios::in);
 while(!fin.eof())
 {
  ch=fin.get();
  cout<<ch;
  if(fin.eof())
   break;
 }
fin.close();
}
void writedata()
{
 char ch='1';
 ofstreamfout("data.txt",ios::out);
 cout<<"\nenter string to be inputed=\ninput 0 to end the string:\n:";
 while(ch!='0')
 {
  cin.get(ch);
  fout.put(ch);
 }
 fout.close();
}
void countalpha()
{
  int x=0;charch;
  ifstream c("data.txt",ios::in);
  while(!c.eof())
 {
  ch=c.get();
  if(isalpha(ch))
  x=x+1;
 }
 cout<<"\nno of alphabets:"<<x;
 c.close();
}
void countnum()
{
 char ch;
 int x=0;
 ifstream c("data.txt",ios::in);
 while(c.eof()==0)
{
  ch=c.get();
  if(isdigit(ch))
  x=x+1;
}
 cout<<"\nno of digit:"<<x-1;
 c.close();
}
void main()
{
  clrscr();
  int x;
  do
 {
   cout<<"\n1.write in a string \n2.read the file \n3.count alpha \n4.count digit
\n5.exit"<<endl;
 cin>>x;
 switch (x)
{
 case 1:writedata();break;
 case 2:readdata();break;
 case 3:countalpha();break;
 case 4:countnum();break;
 case 5:exit(0);break;
}
}while(x!=5);
getch();
}
```



```
1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
1

enter string to be inputed=
input 0 to end the string:
:hello people 4567 qwerty0

1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
2

hello people 4567 qwerty0
1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
_
```



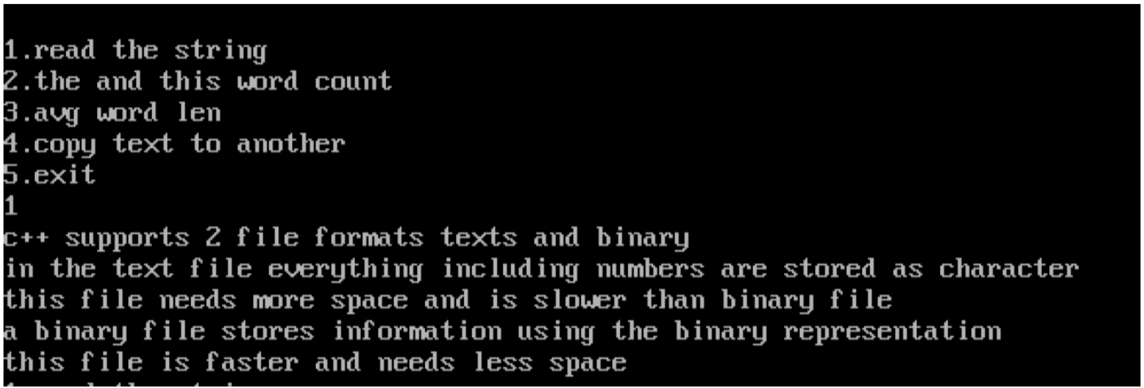
```
1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
3

no of alphabets:17
1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
4

no of digit:4
1.write in a string
2.read the file
3.count alpha
4.count digit
5.exit
5_
```

9. **MAKE A MENU BASED PROGRAM TO PERFORM OPERTIONS ON TEXT FILE (NOTES.TXT) WORD BY WORD**
1-READ THE FILE CONTENTS.
2-COUNT ALL THE WORDS AND OCCURENCES OF 'THIS' AND 'THE'.
3-FIND THE AVERAGE WORD LENGTH OF THE FILE.
4-COPY THE CONTENTS TO ANOTHER FILE MYNOTES.TXT WORD BY WORD EITHER BEGGINNING WITH 'T'/'t'OR ENDING WITH 'e'.

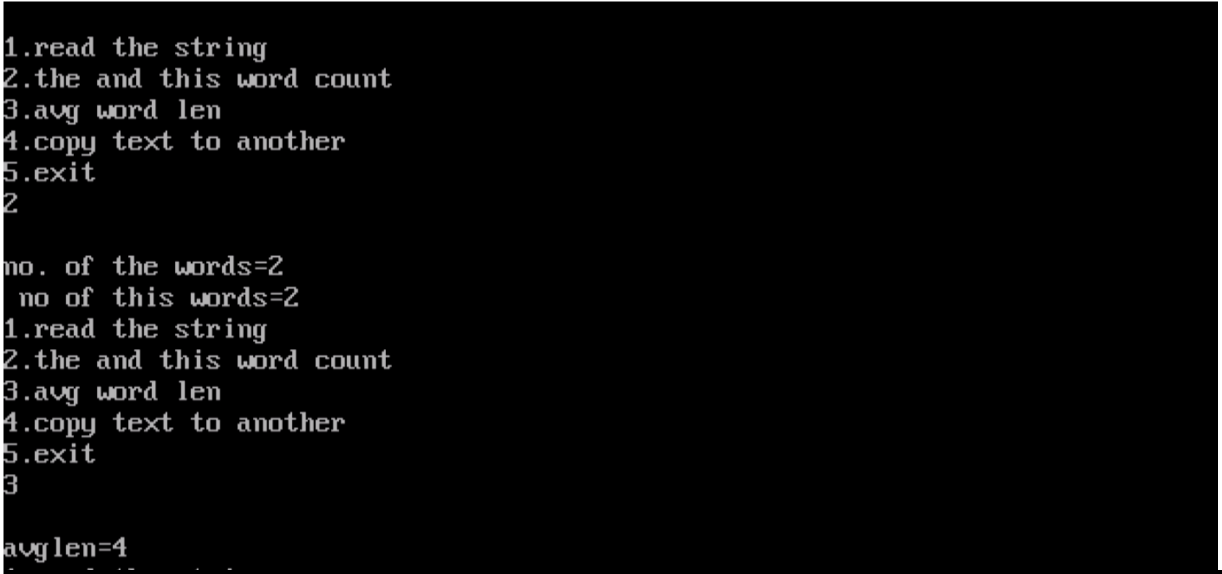
```
#include<fstream.h>
#include<conio.h>
#include<string.h>
#include<process.h>
void readdata()
{
 char ch;
ifstream fin("notes.txt",ios::in);
 while(!fin.eof())
 {
ch=fin.get();
cout<<ch;
  if(fin.eof())
  break;
 }
```

```
fin.close();
}
void count()
{
ifstream fin("notes.txt");
 int x=0,y=0;
 char ch[50];
 while(!fin.eof())
 {
  fin>>ch;
  if(strcmpi(ch,"the")==0)
  {
   x++;
  }
  else if(strcmpi(ch,"this")==0)
  {
   y++;
   }
  if(fin.eof())
 break;
}
cout<<"\nno. of the words="<<x<<"\n no of this words="<<y;
}
void avglen()
{
ifstream fin("notes.txt");
char word[50];
int x=0,sum=0;
while(fin>>word)
{
x++;
sum+=strlen(word);
if(fin.eof())
break;
}
cout<<"\navglen="<<sum/x;
fin.close();
}
void copy()
{
char ch[25];int y,z=0;
int x;
```

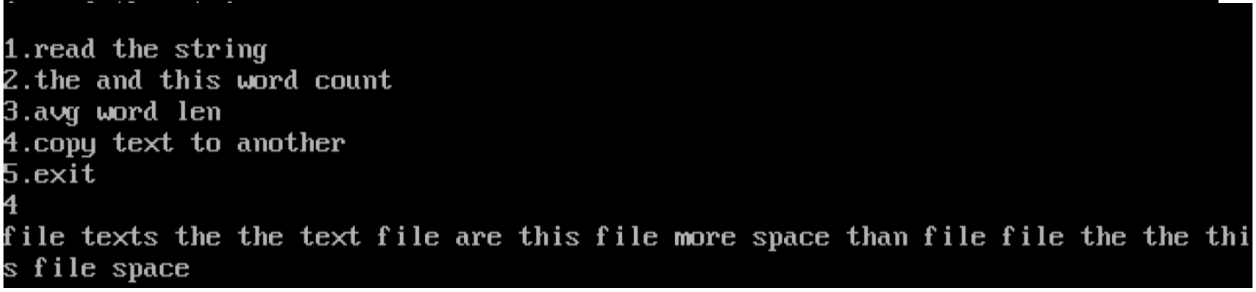
```
ifstream fin("notes.txt");
ofstreamfout("mynotes.txt");
while(fin)
{
  fin>>ch;
  y=strlen(ch);
  if(ch[0]=='t'||ch[0]=='T')
 {
   x++;
   fout<<ch<<" ";
   cout<<ch<<" ";
 }
else if(ch[y-1]=='e')
{
  z++;
  fout<<ch<<" ";
  cout<<ch<<" ";
}
 if (fin.eof())
 break;
 }
}
void main()
{
 clrscr();
 int x;
 do
{
  cout<<"\n1.read the string\n2.the and this word count \n3.avg word len\n4.copy text to
another  \n5.exit"<<endl;
  cin>>x;
 switch (x)
{
  case 1:readdata();break;
  case 2:count();break;
  case 3:avglen();break;
  case 4:copy();break;
  case 5:exit(0);break;
 }
}while(x!=5);
getch();
}
```



```
1.read the string
2.the and this word count
3.avg word len
4.copy text to another
5.exit
1
c++ supports 2 file formats texts and binary
in the text file everything including numbers are stored as character
this file needs more space and is slower than binary file
a binary file stores information using the binary representation
this file is faster and needs less space
```



```
1.read the string
2.the and this word count
3.avg word len
4.copy text to another
5.exit
2

no. of the words=2
 no of this words=2
1.read the string
2.the and this word count
3.avg word len
4.copy text to another
5.exit
3

avglen=4
```



```
1.read the string
2.the and this word count
3.avg word len
4.copy text to another
5.exit
4
file texts the the text file are this file more space than file file the the thi
s file space
```

10 **MAKE A MENU BASED PROGRAM TO PERFORM THE FOLLOWING OPERATIONS ON TEXT FILE(FILE.TXT) LINE BY LINE**

1-READ

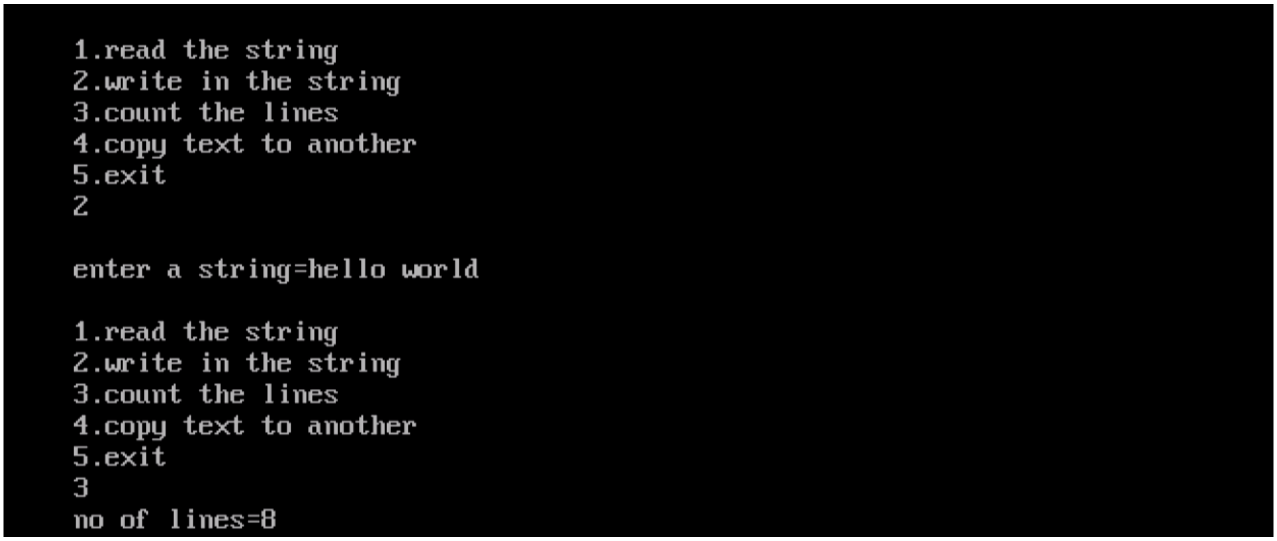
2-COUNT LINES

3-WRITE LINES IN A FILE

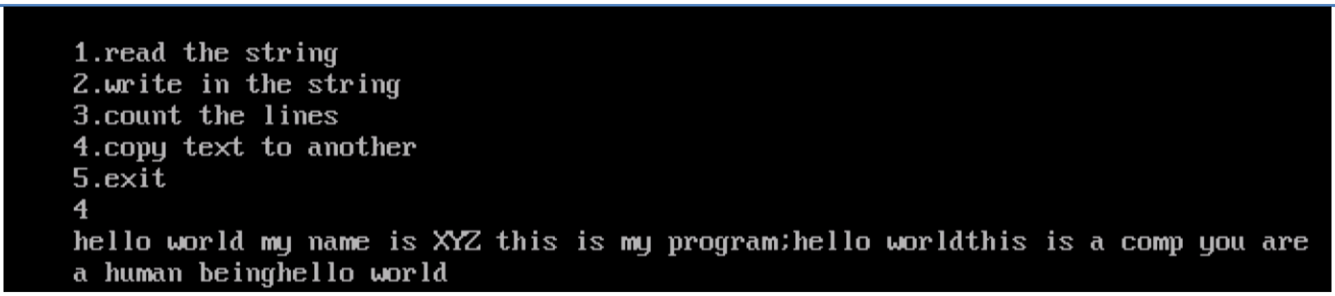
4-COPY ONLY THOSE LINES TO MYFILE.TXT WHICH BEGIN WITH A VOWEL

```
#include<fstream.h>
#include<conio.h>
#include<process.h>
#include<stdio.h>
void read()
{
```

```
char line[200];
ifstream fin("fin.txt");
while(fin.get line(line,200,'.')
{
 cout<<line<<endl;
 if(fin.eof())
 break;
}
fin.close();
}
void copy()
{
char line[200];
ifstream fin("file.txt");
ofstreamfout("myfile.txt");
while(fin.getline(line,200,'.')
{
 line[0]=tolower(line[0]);
 if(line[0]=='a'||line[0]=='e'||line[0]=='o'||line[0]=='u')
 fout<<line<<endl;
 if(fin.eof())
 break;
 }
fin.close();
}
void write()
{
 char line[200];
 ofstreamfout("file.txt",ios::app);
 cout<<"\nenter a string=";
 gets(line);
 fout<<line<<";";
 fout.close();
}
void count()
{
char line[200];
int x=0;
while(fin.getline(line,200,';')
{
  x++;
}
cout<<"no of lines="<<x;
}
void main()
{
clrscr();
int x;
do
{
 cout<<"\n1.read the string\n2.write in the string \n3.count the lines \n4.copy text to another  \n5.exit"<<endl;
  cin>>x;
switch (x)
{
  case 1:read();break;
  case 3:count();break;
  case 2:write();break;
  case 4:copy();break;
  case 5:exit(0);break;
}
}while(x!=5);
getch();
}
```



```
1.read the string
2.write in the string
3.count the lines
4.copy text to another
5.exit
2

enter a string=hello world

1.read the string
2.write in the string
3.count the lines
4.copy text to another
5.exit
3
no of lines=8
```

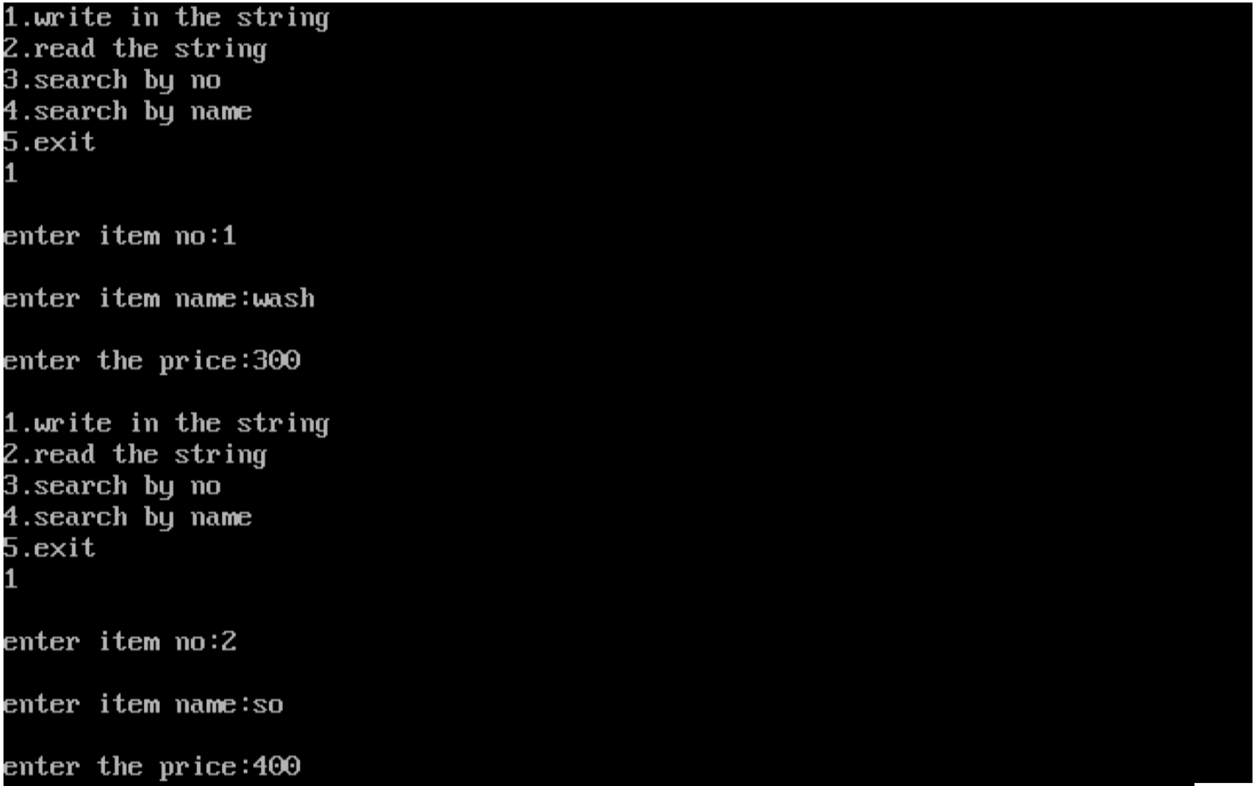


```
1.read the string
2.write in the string
3.count the lines
4.copy text to another
5.exit
4
hello world my name is XYZ this is my program;hello worldthis is a comp you are
a human beinghello world
```

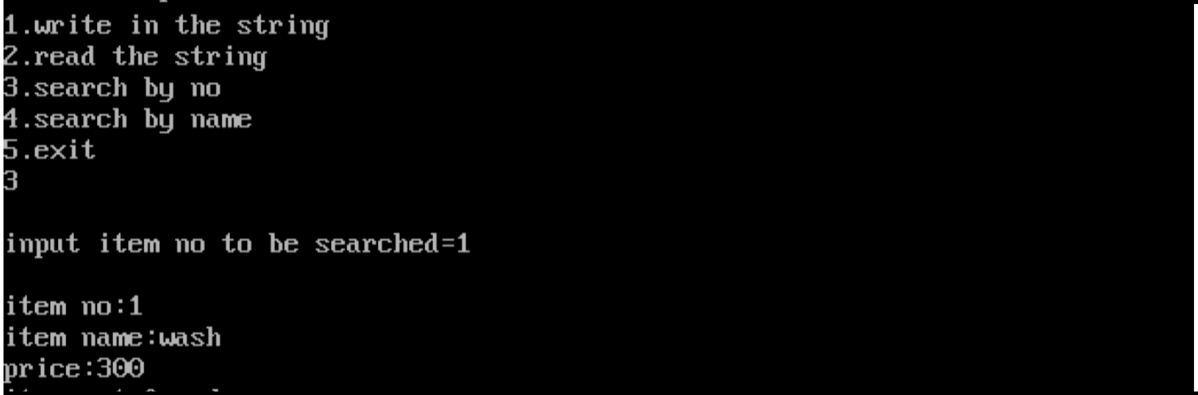
11 MAKE A MENU BASED PROGRAM TO PERFORM FOLLOWING OPERATIONS ON A BINARY FILE ITEM.DAT
1-READ THE CONTENTS
2-WRITE THE CONTENTS
3-SEARCH THE CONTENTS
FILE CONTAINS OBJECT OF CLASS ITEM

```
#include<fstream.h>
#include<conio.h>
#include<process.h>
#include<string.h>
#include<stdio.h>
class item
{
int itemno;
char iname[30];
float price;
int qty;
public:void getdata()
        {
        cout<<"\nenter item no:";
        cin>>itemno;
        cout<<"\nenter item name:";
        gets(iname);
        cout<<"\nenter the price:";
        cin>>price;
        }
        void showdata()
        {
        cout<<"\nitem no:"<<itemno<<"\nitem name:"<<iname<<"\nprice:"<<price;
        }
        int getitemno()
        {
           return itemno;
        }
        char* getiname()
        {
           return iname;
        }
 };
 void writefile()
 {
  item i;
  ofstream fout("sitem.dat",ios::binary|ios::app);
  i.getdata();
  fout.write((char*)&i,sizeof(i));
  fout.close();
 }
 void readfile()
 {
  item i;
  ifstream fin("sitem.dat",ios::binary|ios::in);
  while(fin.read((char*)&i,sizeof(i)))
 {
  i.showdata();
  if(fin.eof())
  break;
 }
 fin.close();
 }
 void searchbyno()
 {
  ifstream fin("sitem.dat",ios::binary|ios::in);
  int no;
  cout<<"\ninput item no to be searched=";
  cin>>no;
  item i;
  while (!fin.eof())
 {
   fin.read((char*)&i,sizeof(i));
   if(i.getitemno()==no)
   i.showdata();
  else
   cout<<"\nitem not found";
  if(fin.eof())
  break;
```

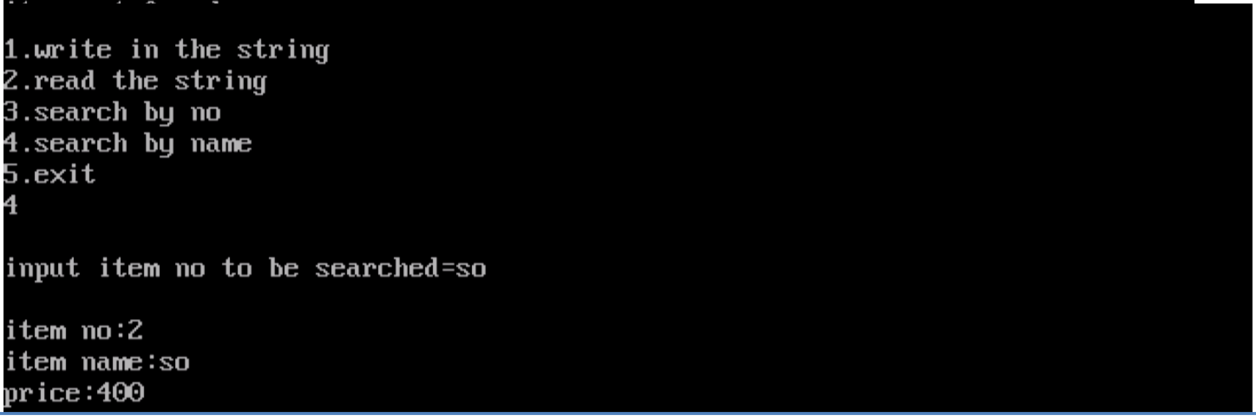
```
}
fin.close();
}
void searchbyname()
{
  ifstream fin("sitem.dat",ios::binary|ios::in);
  char ch[20],name[20];
  cout<<"\ninput item no to be searched=";
  gets(ch);
  item i;
  int count=0;
   while (!fin.eof())
   {
      fin.read((char*)&i,sizeof(i));
      strcpy(name,i.getiname());
      if(strcmpi(name,ch)==0)
      {
        i.showdata();
         count=1;
      }
     if(fin.eof())
     break;
  }
 if(count==0)
cout<<"\nitem not found";
fin.close();
}
void main()
{
clrscr();
int x;
do
{
 cout<<"\n1.write in the string\n2.read the string \n3.search by no \n4.search by name
\n5.exit"<<endl;
 cin>>x;
 switch (x)
 {
  case 2:readfile();break;
  case 3:searchbyno();break;
  case 1:writefile();break;
  case 4:searchbyname();break;
  case 5:exit(0);break;
}
}while(x!=5);
getch();
}
```



```
1.write in the string
2.read the string
3.search by no
4.search by name
5.exit
1

enter item no:1

enter item name:wash

enter the price:300

1.write in the string
2.read the string
3.search by no
4.search by name
5.exit
1

enter item no:2

enter item name:so

enter the price:400
```



```
1.write in the string
2.read the string
3.search by no
4.search by name
5.exit
3

input item no to be searched=1

item no:1
item name:wash
price:300
```



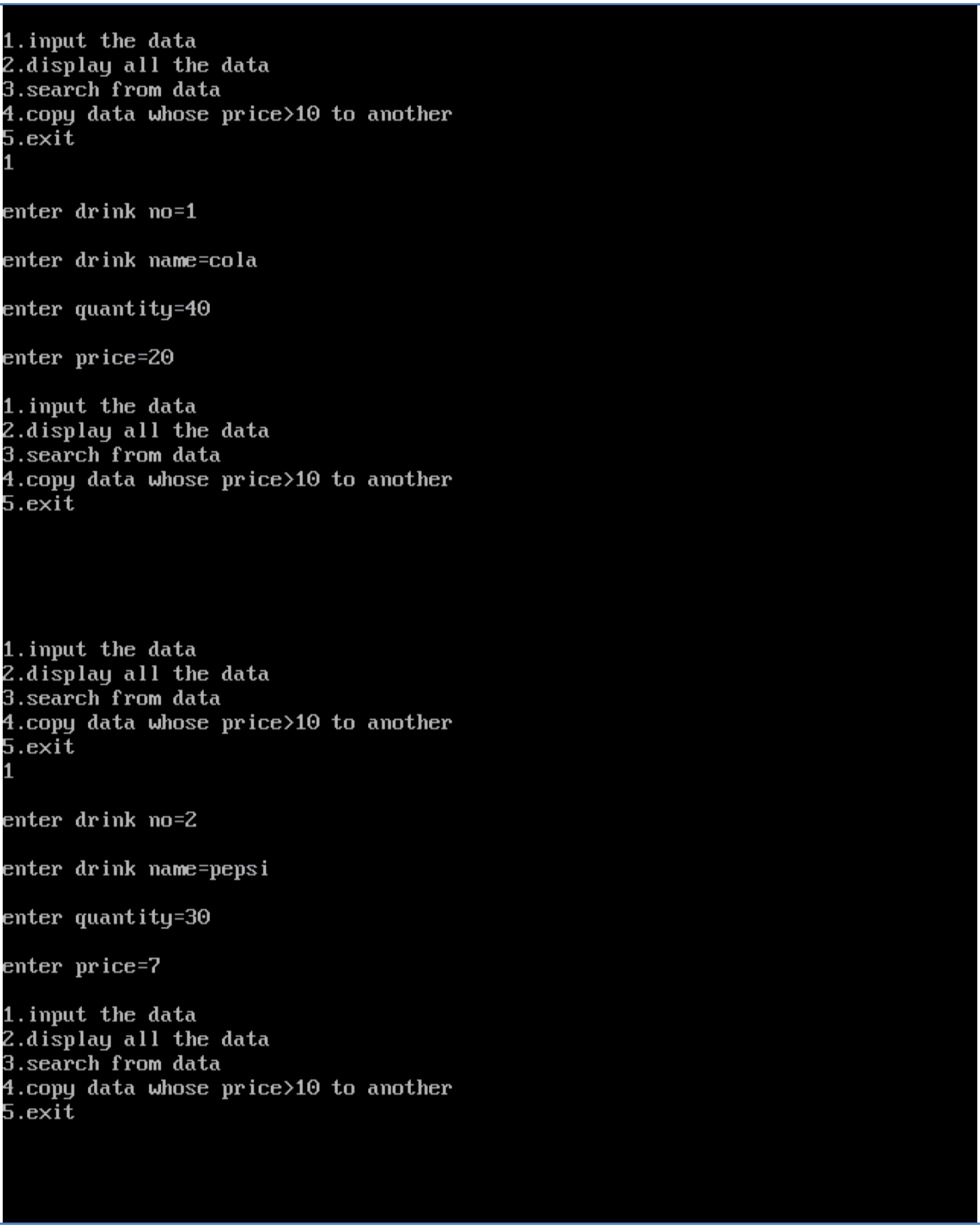
```
1.write in the string
2.read the string
3.search by no
4.search by name
5.exit
4

input item no to be searched=so

item no:2
item name:so
price:400
```

12 **MAKE A MENU BASED PROGRAM TO PERFORM THE FOLLOWING OPERATIONS ON A BINARY FILE DRINK.DAT**
1-READ THE CONTENS
2-WRITE THE CONTENTS
3-SEARCH THE CONTENT
4- COPY ONLY THOSE RECORDS WHOSE PRICE IS MORE THAN 10 TO MYDRINK.DAT. ASLO DISPLAY CONTENTS OF MYDRINKS.DAT
FILE CONTAINS OBJECT OF CLASS DRINK

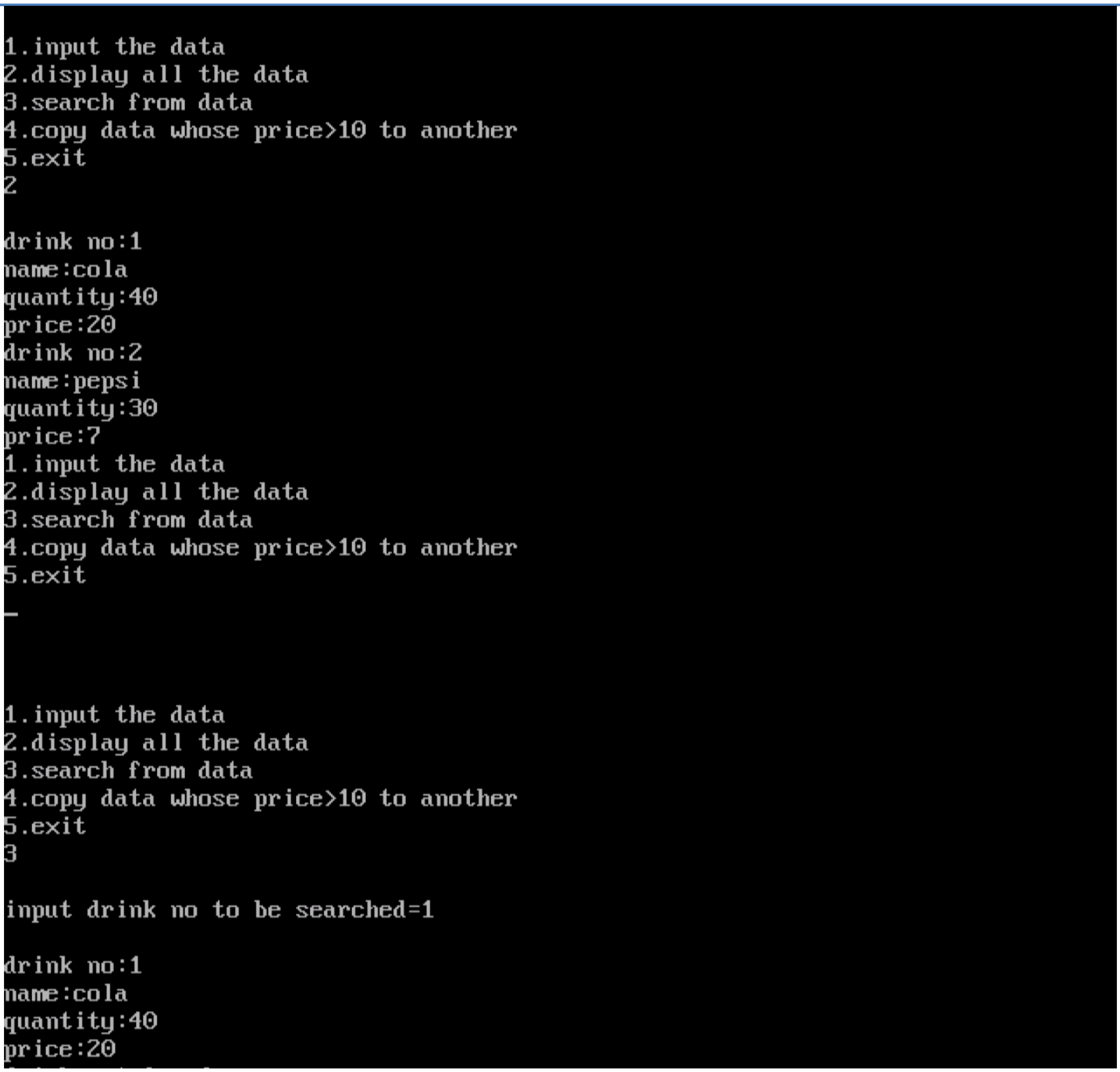
```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
class mydrink
{
int drinkno;
char dname[20];
int qty;
float price;
public:void input()
        {
cout<<"\nenter drink no=";
cin>>drinkno;
cout<<"\nenter drink name=";
cin>>dname;
cout<<"\nenter quantity=";
cin>>qty;
cout<<"\nenter price=";
cin>>price;
        }
        void show()
        {
cout<<"\ndrink
no:"<<drinkno<<"\nname:"<<dname<<"\nquantity:"<<qty<<"\nprice:"<<price;
        }
        int getprice()
        {
           return price;
        }
        int getdrinkno()
```

```
      {
      return drinkno;
      }
};
void writefile()
{
  mydrinki;
  ofstreamfout("drink.dat",ios::binary|ios::app);
  i.input();
  fout.write((char*)&i,sizeof(i));
  fout.close();
}
void readfile()
{
  mydrinki;
  ifstream fin("drink.dat",ios::binary|ios::in);
  while(fin.read((char*)&i,sizeof(i)))
 {
  i.show();
  if(fin.eof())
  break;
 }
fin.close();
}
void searchbyno()
{
 ifstream fin("drink.dat",ios::binary|ios::in);
 int no;
 cout<<"\ninput drink no to be searched=";
 cin>>no;
 mydrinki;
 while (!fin.eof())
{
  fin.read((char*)&i,sizeof(i));
 if(i.getdrinkno()==no)
 i.show();
 else
  cout<<"\ndrink not found";
 if(fin.eof())
 break;
}
fin.close();
```

```
}
void copy()
{
 mydrinki;
 ifstream fin("drink.txt");
 ofstreamfout("mydrink.txt");
 while(fin.eof()==0)
{
  fin.read((char*)&i,sizeof(i));
  if(i.getprice()>10)
  {
   fout.write((char*)&i,sizeof(i));
   i.show();
 }
if(fin.eof())
break;
}
fin.close();
}
void main()
{
 clrscr();
 int x;
 do
{
  cout<<"\n1.input the data\n2.display all the data \n3.search from data \n4.copy data
whose price>10 to another  \n5.exit"<<endl;
  cin>>x;
  switch (x)
 {
   case 1:writefile();break;
   case 3:searchbyno();break;
   case 2:readfile();break;
   case 4:copy();break;
   case 5:exit(0);break;
 }
}while(x!=5);
getch();
}
```



```
1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
1

enter drink no=1

enter drink name=cola

enter quantity=40

enter price=20

1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit




1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
1

enter drink no=2

enter drink name=pepsi

enter quantity=30

enter price=7

1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
```



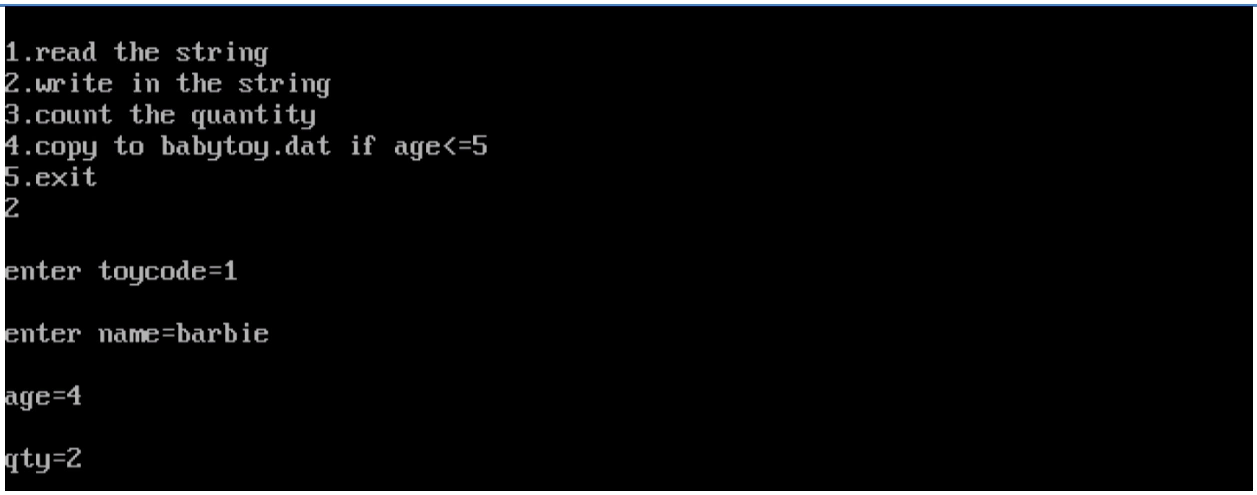
```
1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
2

drink no:1
name:cola
quantity:40
price:20
drink no:2
name:pepsi
quantity:30
price:7
1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
_



1.input the data
2.display all the data
3.search from data
4.copy data whose price>10 to another
5.exit
3

input drink no to be searched=1

drink no:1
name:cola
quantity:40
price:20
```

13 **MAKE A MENU BASED PROGRAM TO PERFORM THE FOLLOWING OPERATIONS ON A BINARY FILE TOY.DAT**
1-READ THE CONTENTS
2-WRITE THE CONTENTS
3-COPY THE CONTENT WHOSE AGE GROUP IS BELOW 5 YRS IN FILE BABYTOY.DAT
ALSO DISPLAY CONTENTS OF THE FILE BABYTOY .DAT IN THE SAME FUNCTION.
4- COUNT AND DISPLAY TOTAL QUANTITY OF TOYS
FILE CONTAINS CLASS OF OBJECT TOY

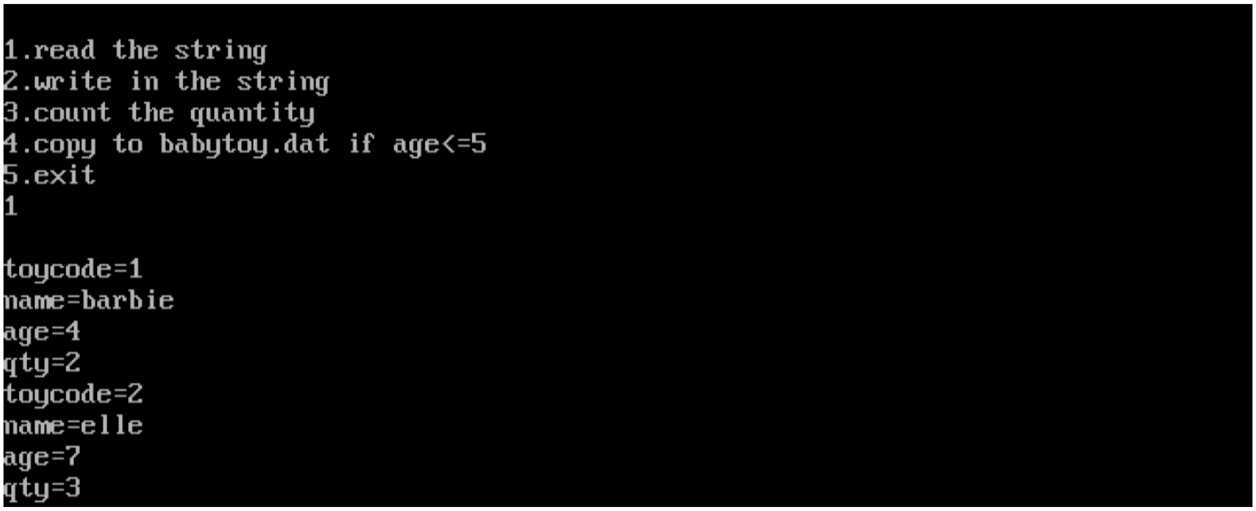
```
#include<fstream.h>
#include<conio.h>
```

```
#include<stdio.h>
#include<process.h>
class toys
{
private:inttoycode;
        char tname[20];
        int age;
        int qty;
public: void enter()
        {
        cout<<"\nentertoycode=";
        cin>>toycode;
        cout<<"\nenter name=";
        gets(tname);
        cout<<"\nage=";
        cin>>age;
        cout<<"\nqty=";
        cin>>qty;
        }
        void show()
        {
        cout<<"\ntoycode="<<toycode<<"\nname="<<tname<<"\nage="<<age<
<"\nqty="<<qty;
        }
        int getage()
        {
        return age;
        }
        int getqty()
        {
        return qty;
        }

};
void writefile()
{
 toys t;
ofstreamfout("toy.dat",ios::binary|ios::app);
t.enter();
fout.write((char*)&t,sizeof(t));
fout.close();
}
void readfile()
{
 toys t;
ifstream fin("toy.dat",ios::binary|ios::in);
 while(fin.read((char*)&t,sizeof(t)))
 {
t.show();
  if(fin.eof())
  break;
 }
fin.close();
}
void copy()
{
 toys t;
 int count=0;
ifstream fin("toy.dat",ios::binary|ios::in);
ofstreamfout("babytoy.dat",ios::binary|ios::out);
 while(fin.read((char*)&t,sizeof(t)))
 {
  if(t.getage()<=5)
  {
fout.write((char*)&t,sizeof(t));
t.show();
  }
 if(fin.eof())
 break;
 }
fin.close();
fout.close();
```

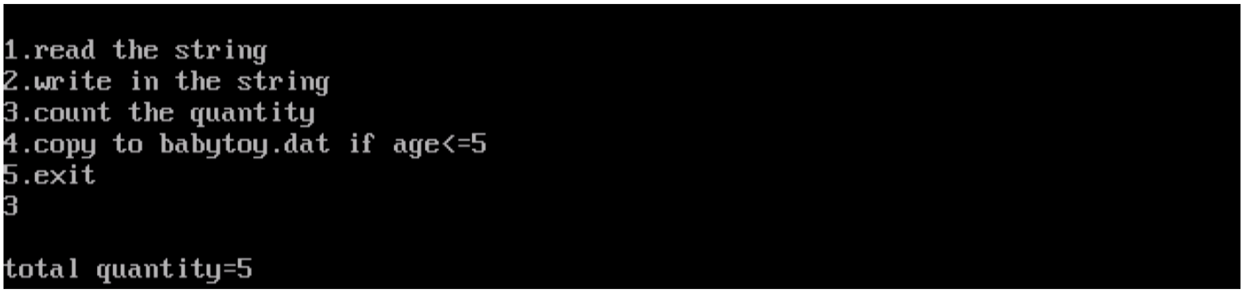
```
}
void count()
{
 toys t;
 int total=0;
ifstream fin("toy.dat",ios::binary);
 while(fin.read((char*)&t,sizeof(t)))
 {
 total+=t.getqty();
 if(fin.eof())
 break;
 }
cout<<"\ntotal quantity="<<total;
fin.close();
 }
void main()
{
clrscr();
int x;
do
{
cout<<"\n1.read the string\n2.write in the string \n3.count the quantity
\n4.copy to babytoy.dat if age<=5  \n5.exit"<<endl;
cin>>x;
 switch (x)
 {
 case 1:readfile();break;
 case 3:count();break;
 case 2:writefile();break;
 case 4:copy();break;
 case 5:exit(0);break;
 }
}while(x!=5);
getch();
}
```



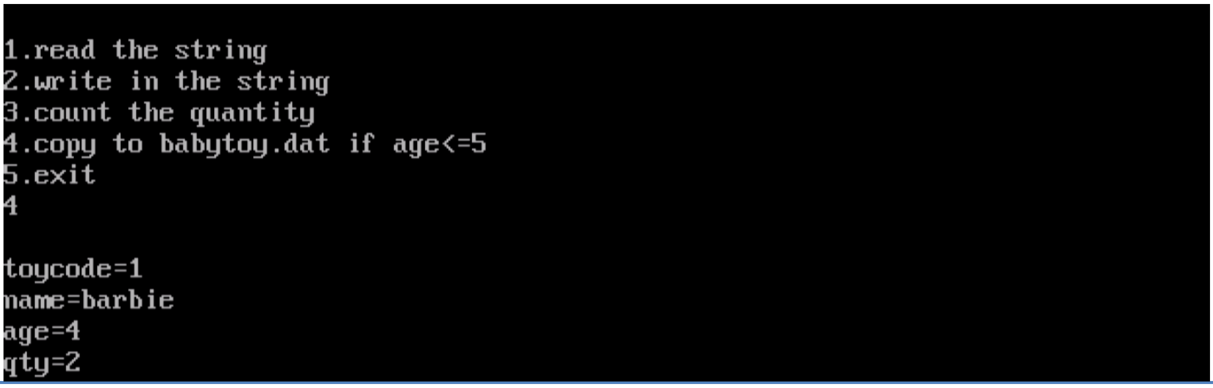
```
1.read the string
2.write in the string
3.count the quantity
4.copy to babytoy.dat if age<=5
5.exit
2

enter toycode=1

enter name=barbie

age=4

qty=2
```



```
1.read the string
2.write in the string
3.count the quantity
4.copy to babytoy.dat if age<=5
5.exit
1

toycode=1
name=barbie
age=4
qty=2
toycode=2
name=elle
age=7
qty=3
```



```
1.read the string
2.write in the string
3.count the quantity
4.copy to babytoy.dat if age<=5
5.exit
3

total quantity=5
```



```
1.read the string
2.write in the string
3.count the quantity
4.copy to babytoy.dat if age<=5
5.exit
4

toycode=1
name=barbie
age=4
qty=2
```

14 Make a menu based program to perform following operations on a binary file flight.dat

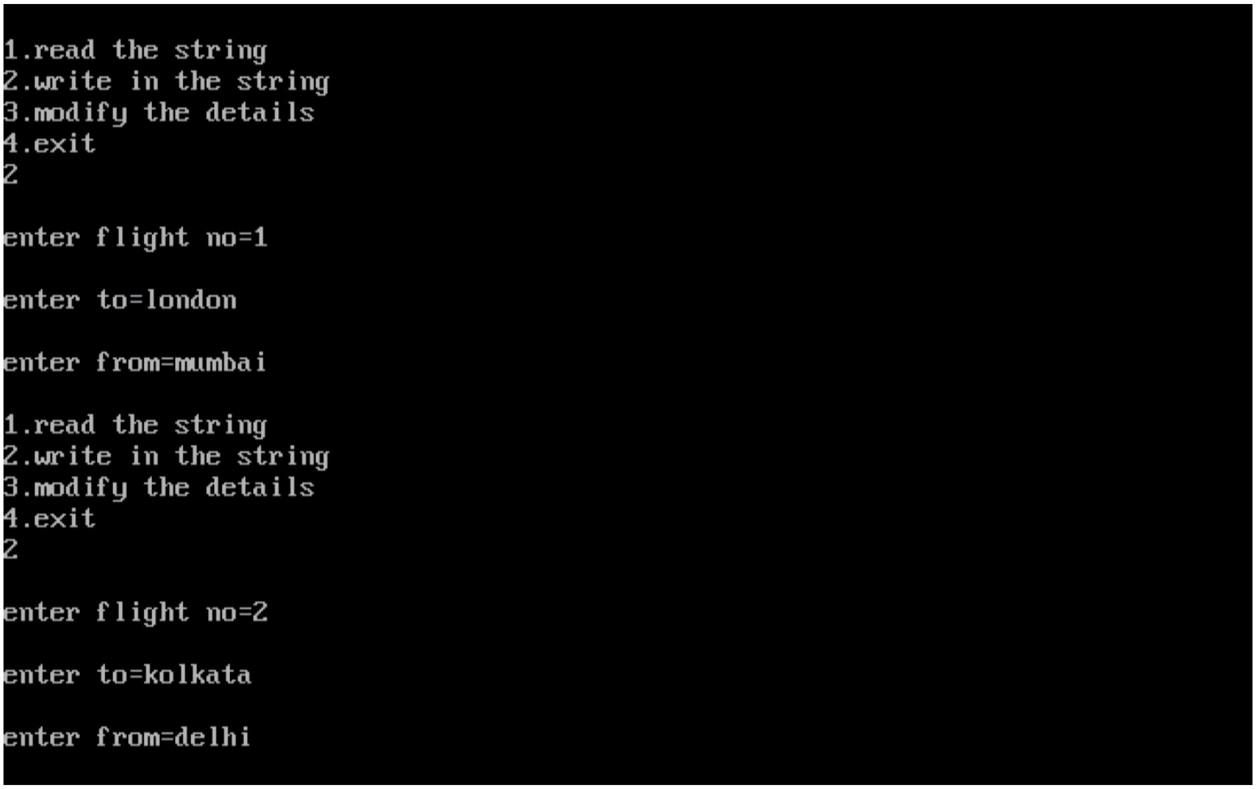
1.Read 2. Write 3. Modify

File contains objects of class flight

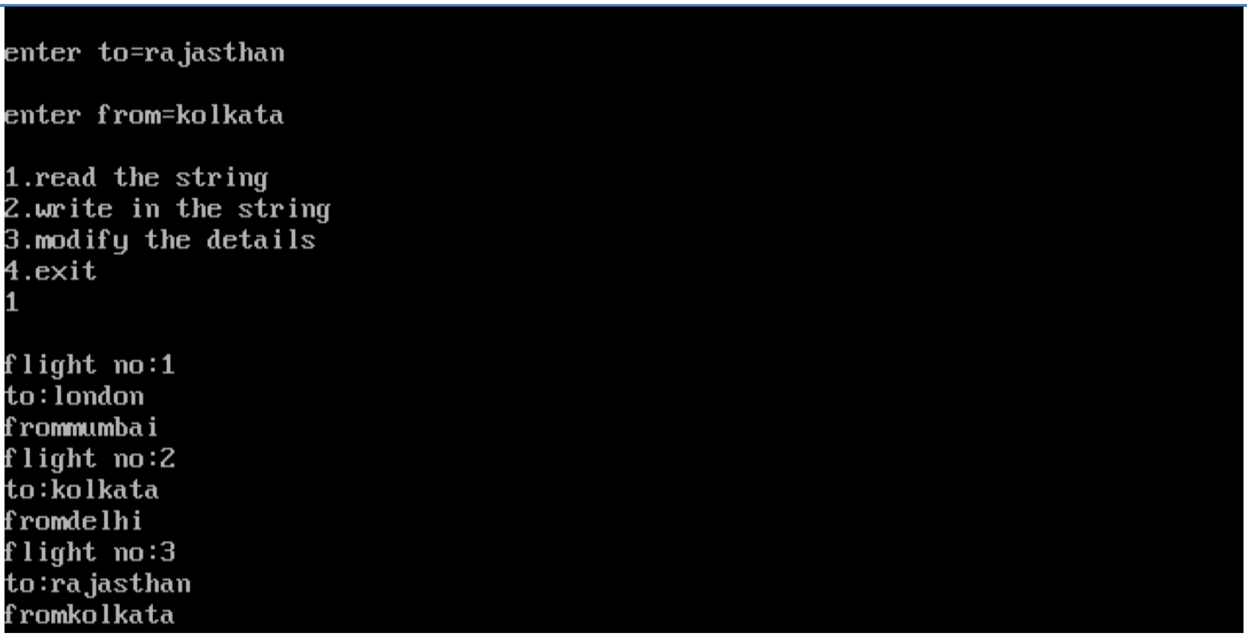
```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<process.h>
class flight
{
 int fno;//flightno
 char from[20];
 char to[20];
 public:
  int getfno()
{
  return fno;
}
void enter()
{
 cout<<"\nenter flight no=";
 cin>>fno;
 cout<<"\nenter to=";
 cin>>to;
 cout<<"\nenter from=";
 cin>>from;
}
void display()
{
 cout<<"\nflight no:"<<fno<<"\nto:"<<to<<"\nfrom"<<from;
}
};
void writefile()
{
 flight f;
 ofstreamfout("flight.dat",ios::binary|ios::app);
 f.enter();
 fout.write((char*)&f,sizeof(f));
 fout.close();
}
```

```
void readfile()
{
 flight f;
 ifstream fin("flight.dat",ios::binary|ios::in);
  while(fin.read((char*)&f,sizeof(f)))
  {
   f.display();
   if(fin.eof())
   break;
 }
fin.close();
}
void modify()
{
int x=0;
flight f,newf;
ifstream file("flight.dat",ios::binary|ios::in) ;
ofstreamfout("flight.dat",ios::binary|ios::ate);
int fno;
int pos;
cout<<"\ninput flight no to be modified:";
cin>>fno;
cout<<"\nenter your records:";
newf.enter();
while(file.read((char*)&f,sizeof(f)))
{
  if (f.getfno()==fno)
  {
    x=1;
    pos = (file.tellg()-sizeof(f));
    fout.seekp( pos,ios::beg);
    fout.write((char*)&newf,sizeof(newf));
  }
 if(file.eof())
 break;
}
if(x==0)
cout<<"\nrecord not found";
file.close();
}
void main()
{
```

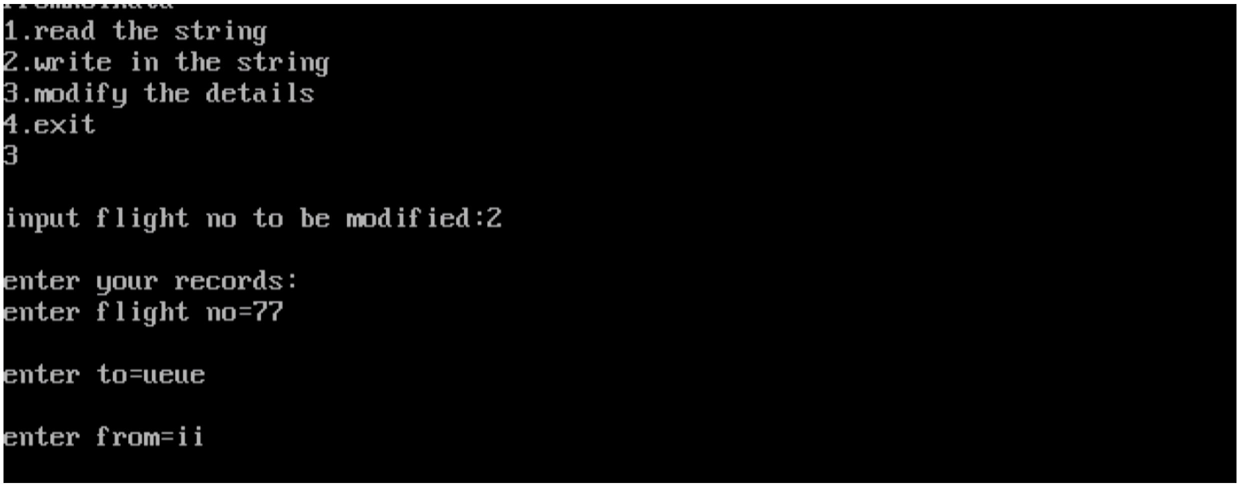
```
clrscr();
int x;
do
{
 cout<<"\n1.read the string\n2.write in the string \n3.modify the details\n4.exit"<<endl;
 cin>>x;
 switch (x)
 {
  case 1:readfile();break;
  case 3:modify();break;
  case 2:writefile();break;
  case 4:exit(0);break;
 }
}while(x!=4);
getch();
}
```



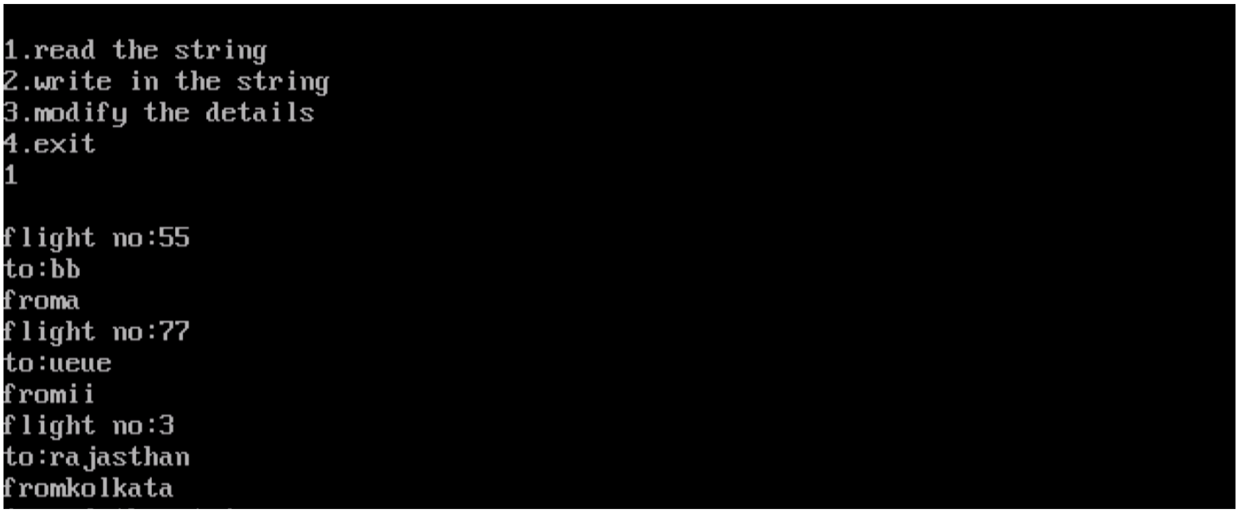
```
1.read the string
2.write in the string
3.modify the details
4.exit
2

enter flight no=1

enter to=london

enter from=mumbai

1.read the string
2.write in the string
3.modify the details
4.exit
2

enter flight no=2

enter to=kolkata

enter from=delhi
```



```
enter to=rajasthan

enter from=kolkata

1.read the string
2.write in the string
3.modify the details
4.exit
1

flight no:1
to:london
frommumbai
flight no:2
to:kolkata
fromdelhi
flight no:3
to:rajasthan
fromkolkata
```



```
1.read the string
2.write in the string
3.modify the details
4.exit
3

input flight no to be modified:2

enter your records:
enter flight no=77

enter to=ueue

enter from=ii
```

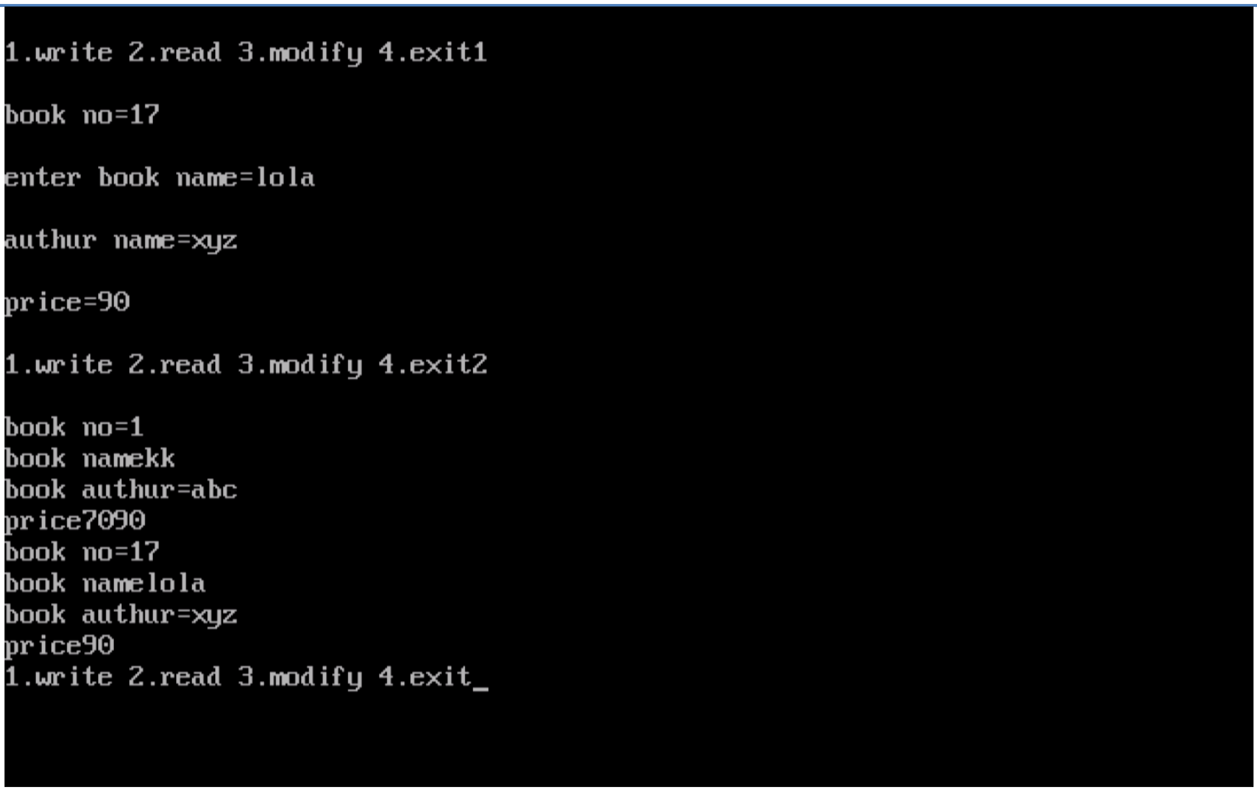


```
1.read the string
2.write in the string
3.modify the details
4.exit
1

flight no:55
to:bb
froma
flight no:77
to:ueue
fromii
flight no:3
to:rajasthan
fromkolkata
```

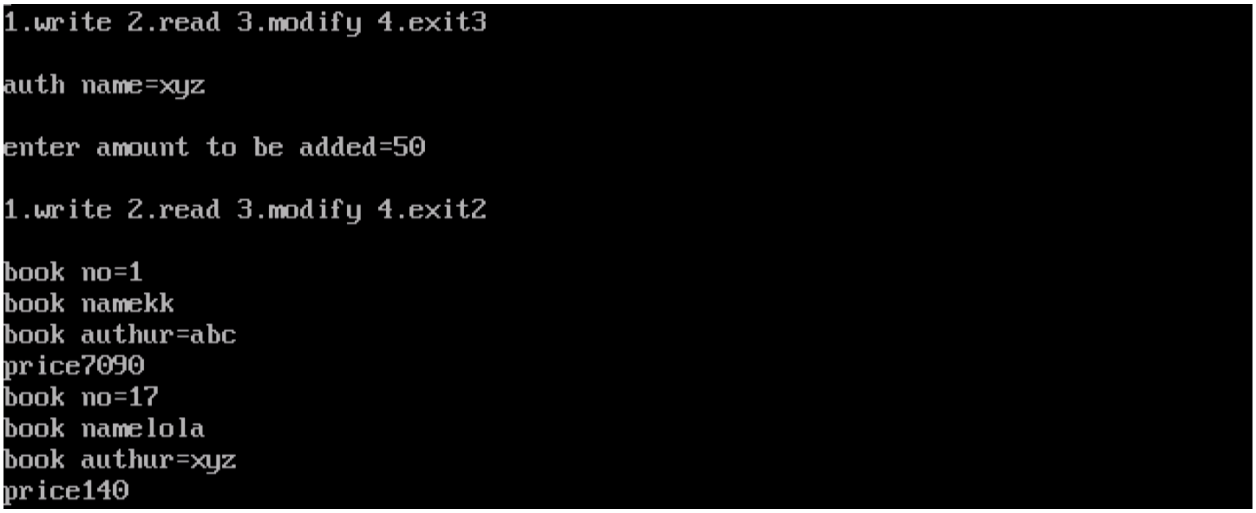
15 **Make a menu based program to perform following operations on a binary file book.dat**
1.Read 2. Write 3. Modify
File contains objects of class book

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<process.h>
class book
{
int bno;
char bname[50],autname[50];
float price;
public:
  void enter()
  {
cout<<"\nbook no=";
cin>>bno;
cout<<"\nenter book name=";
cin>>bname;
cout<<"\nauthur name=";
cin>>autname;
cout<<"\nprice=";
cin>>price;
 }
  void display()
  {
 cout<<"\nbook no="<<bno<<"\nbookname"<<bname<<"\nbookauthur="<<autname<<
"\nprice"<<price;
    }
    char* authname()
    {
    return autname;
    }
    void update(int n)
    {
    price=price+n;
    }
};
```

```
void readfile()
{
 book b;
 ifstream fin("book.dat",ios::binary|ios::in);
 while(fin.read((char*)&b,sizeof(b)))
 {
  b.display();
  if(fin.eof())
  break;
 }
fin.close();
}
void writefile()
{
  book b;
  ofstreamfout("book.dat",ios::binary|ios::app);
  b.enter();
  fout.write((char*)&b,sizeof(b));
  fout.close();
 }
void modify()
{
  int n;
  book b;
  ifstream file("book.dat",ios::binary|ios::in) ;
  ofstreamfout("book.dat",ios::binary|ios::ate);
  char aut[50];
  cout<<"\nauth name=";
  cin>>aut;
  int x=0,pos;
  while(file.read((char*)&b,sizeof(b)))
 {
  if (strcmpi(b.authname(),aut)==0)
 {
   x=1;
   cout<<"\nenter amount to be added=";
   cin>>n;
   b.update(n);
   pos = (file.tellg()-sizeof(b));
   fout.seekp( pos,ios::beg);
   fout.write((char*)&b,sizeof(b));
 }
 if(file.eof())
 break;
}
if(x==0)
 cout<<"\nrecord not found";
 file.close();
}
void main()
{
clrscr();
int menu;
do
{
 cout<<"\n1.write 2.read 3.modify 4.exit";
 cin>>menu;
 switch(menu)
 {
  case 1:writefile();break;
  case 2:readfile();break;
  case 3:modify();break;
  case 4:exit(0);break;
 }
}while(menu!=4);
getch();
}
```



```
1.write 2.read 3.modify 4.exit1

book no=17

enter book name=lola

authur name=xyz

price=90

1.write 2.read 3.modify 4.exit2

book no=1
book namekk
book authur=abc
price7090
book no=17
book namelola
book authur=xyz
price90
1.write 2.read 3.modify 4.exit_
```



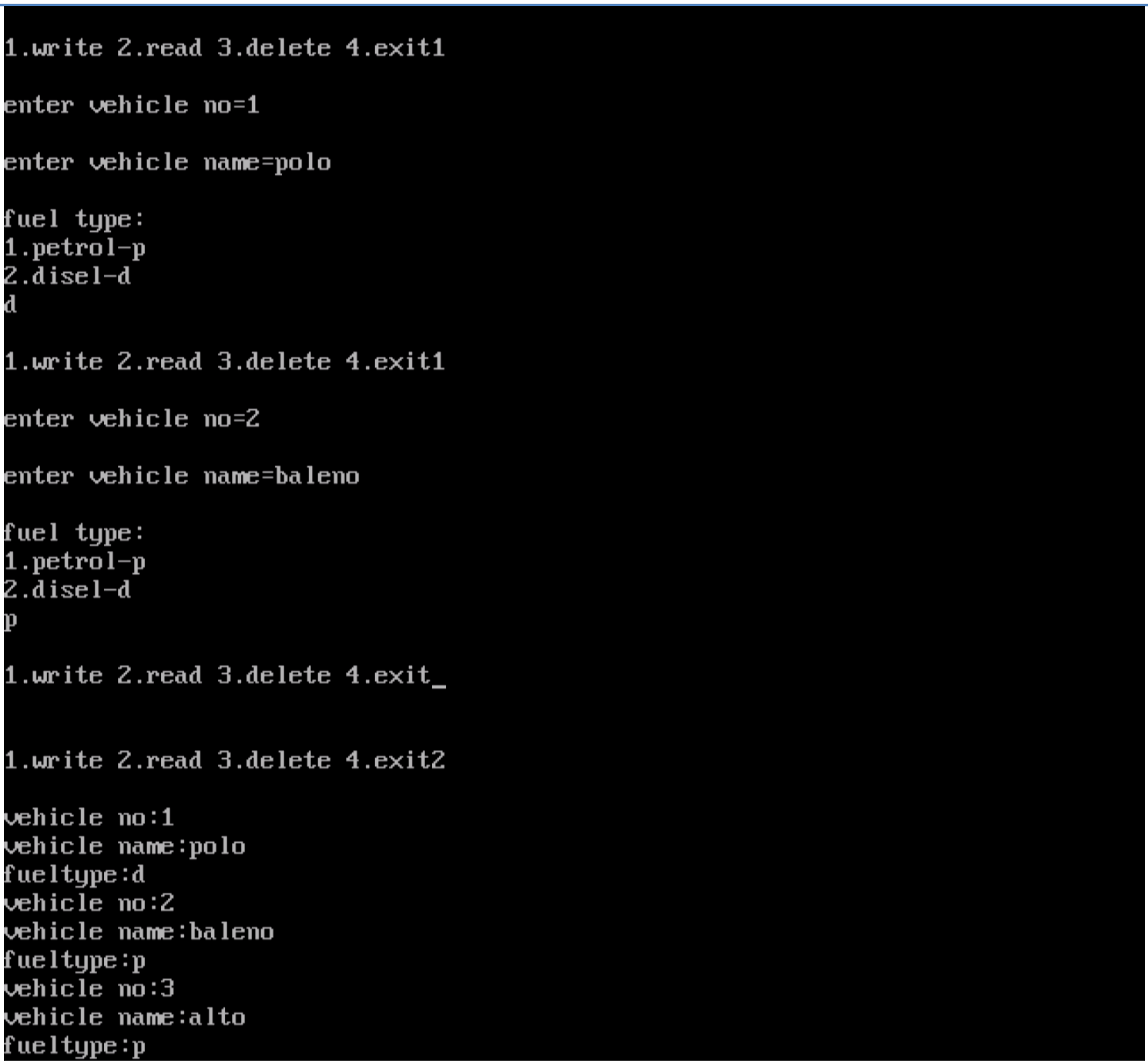
```
1.write 2.read 3.modify 4.exit3

auth name=xyz

enter amount to be added=50

1.write 2.read 3.modify 4.exit2

book no=1
book namekk
book authur=abc
price7090
book no=17
book namelola
book authur=xyz
price140
```

16 Make a menu based program to perform following operations on a binary file vehicle.dat
1.Read 2. Write 3. Delete
File contains objects of class vehicle

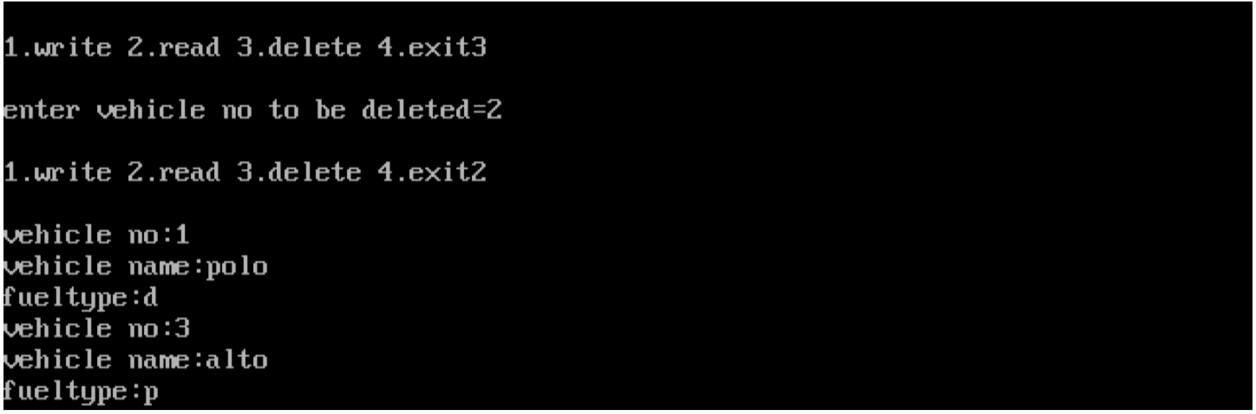
```
#include<fstream.h>
#include<stdio.h>
#include<conio.h>
#include<process.h>
class vehicle
```

```
{
private:intvno;
        char vname[40];
        char fueltype;//petrol p, disel d
public:void input()
      {
        cout<<"\nenter vehicle no=";
        cin>>vno;
        cout<<"\nenter vehicle name=";
        gets(vname);
        cout<<"\nfuel type:\n1.petrol-p\n2.disel-d"<<endl;
        cin>>fueltype;
      }
      void display()
      {
cout<<"\nvehicle no:"<<vno<<"\nvehicle name:"<<vname<<"\nfueltype:"<<fueltype;
      }
      int getvno()
      {
        return vno;
      }
};
void readfile()
{
vehicle v;
ifstream fin("vehicle.dat",ios::binary|ios::in);
while(fin.read((char*)&v,sizeof(v)))
{
v.display();
if(fin.eof())
break;
}
fin.close();
}
void writefile()
{
vehicle v;
ofstreamfout("vehicle.dat",ios::binary|ios::app);
v.input();
fout.write((char*)&v,sizeof(v));
fout.close();
}
```

```
void deletE()
{
vehicle v;
ifstream fin("vehicle.dat",ios::binary|ios::in);
int x=0,no;
ofstreamfout("temp.dat",ios::binary|ios::app);
cout<<"\nenter vehicle no to be deleted=";
cin>>no;
while(fin.read((char*)&v,sizeof(v)))
{
  if(v.getvno()!=no)
  fout.write((char*)&v,sizeof(v));
  else
  x=1;
 if(fin.eof())
 break;
}
if(x==0)
 cout<<"record not found";
 fin.close();
fout.close();
remove("vehicle.dat");
rename("temp.dat","vehicle.dat");
}
void main()
{
clrscr();
int menu;
do
{
 cout<<"\n1.write 2.read 3.delete 4.exit";
 cin>>menu;
 switch(menu)
{
 case 1:writefile();break;
 case 2:readfile();break;
 case 3:deletE();break;
 case 4:exit(0);break;
}
}while(menu!=4);
getch();
}
```



```
1.write 2.read 3.delete 4.exit1

enter vehicle no=1

enter vehicle name=polo

fuel type:
1.petrol-p
2.disel-d
d

1.write 2.read 3.delete 4.exit1

enter vehicle no=2

enter vehicle name=baleno

fuel type:
1.petrol-p
2.disel-d
p

1.write 2.read 3.delete 4.exit_


1.write 2.read 3.delete 4.exit2

vehicle no:1
vehicle name:polo
fueltype:d
vehicle no:2
vehicle name:baleno
fueltype:p
vehicle no:3
vehicle name:alto
fueltype:p
```



```
1.write 2.read 3.delete 4.exit3

enter vehicle no to be deleted=2

1.write 2.read 3.delete 4.exit2

vehicle no:1
vehicle name:polo
fueltype:d
vehicle no:3
vehicle name:alto
fueltype:p
```

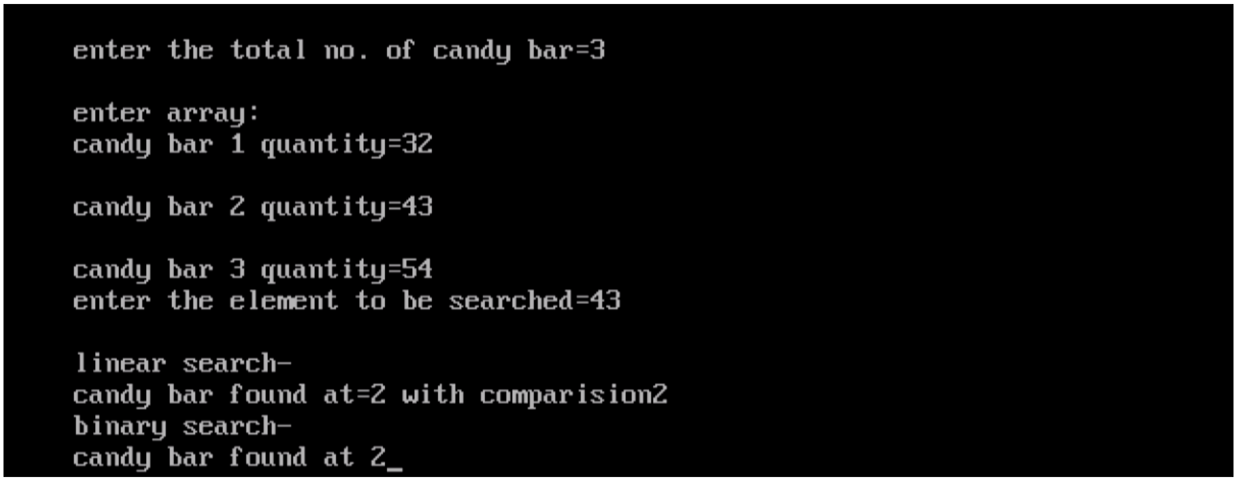
17 Make a menu based program using function to implement following operations on array of 5 records of CandyBar. Input records in sorted order.

1. Linear search

2. Binary Search

3. Display

```
#include<iostream.h>
#include<conio.h>
void lsearch(int arr[],int size,int ele)
{
 int flag=0,comp=1;
 for (int x=0;x<size;x++)
 {
  if(ele==arr[x])
  {
   cout<<"\nlinear search-";
   cout<<"\ncandy bar found at="<<x+1<<" with comparision"<<comp;
   flag++;break;
  }
 else comp++;
}
if(flag==0)
cout<<"\nnot found";
}
void bsearch(int a[],int size, int ele)
{
 int flag,beg, end, mid;
 beg=0; end=size-1;flag=0;
 while(beg<end)
 {
  mid=(beg+end)/2;
  if(ele==a[mid])
  {
   cout<<"\nbinary search-";
   cout<<"\ncandy bar found at "<<mid+1;
   flag=1; break;
  }
  else if(ele<a[mid])
  end=mid-1;
  else if(ele>a[mid])
  end=mid+1;
 }
 if(flag==0)   cout<<"\nnot found";
}
void main()
{
clrscr();
int size, arr[20], ele;
cout<<"\nenter the total no. of candy bar=";
cin>>size;
cout<<"\nenter array:";
for(int x=0;x<size;x++)
{
 cout<<"\ncandy bar "<<x+1<<" quantity=";
 cin>>arr[x];
}
cout<<"enter the element to be searched=";
cin>>ele;
lsearch(arr,size,ele);
bsearch(arr,size,ele);
getch();
}
```



```
enter the total no. of candy bar=3

enter array:
candy bar 1 quantity=32

candy bar 2 quantity=43

candy bar 3 quantity=54
enter the element to be searched=43

linear search-
candy bar found at=2 with comparision2
binary search-
candy bar found at 2_
```

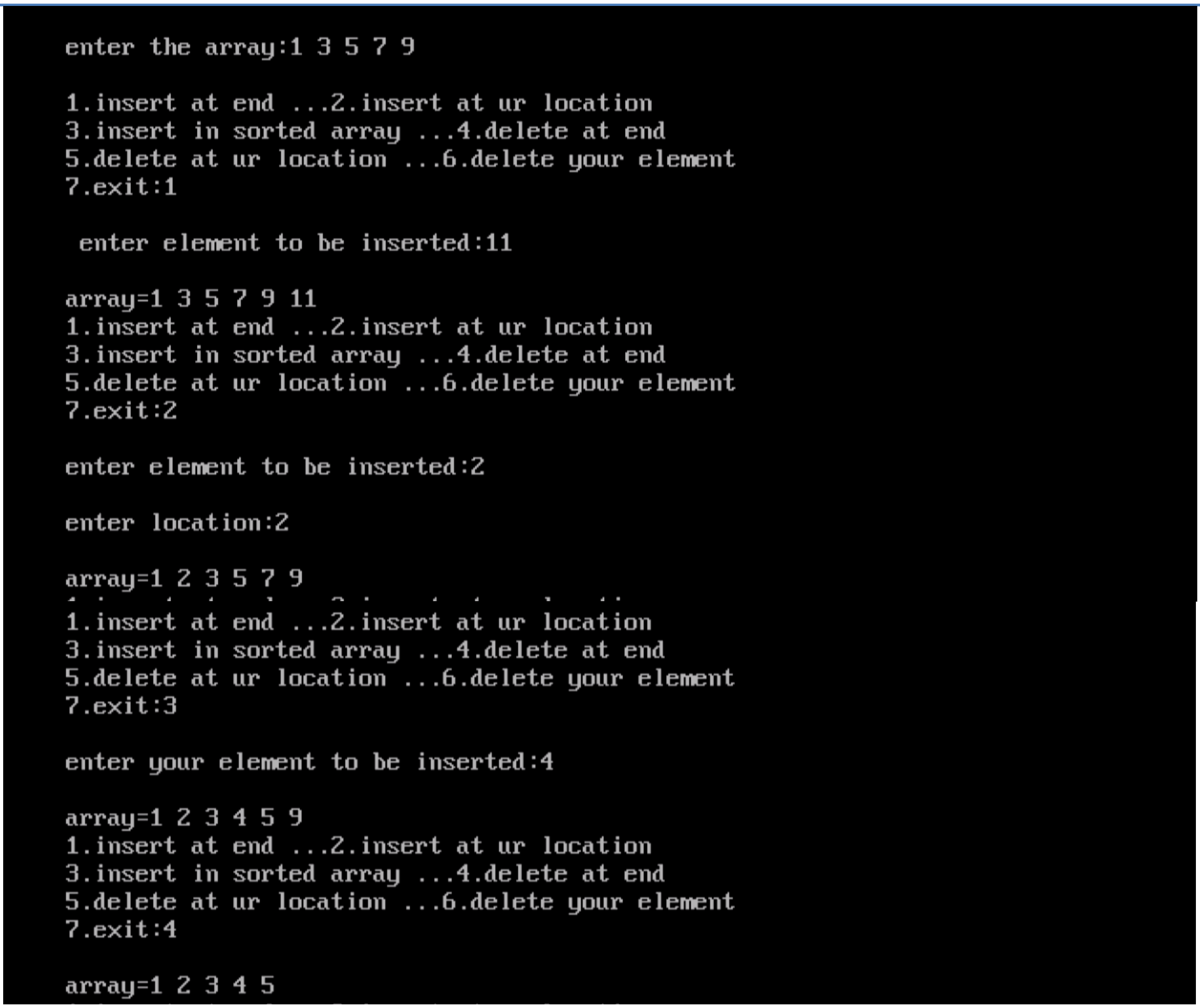
18 Program to implement following operations on 1D array

1. **Insertion at end**
2. **Insertion at given location**
3. **Insertion in sorted array**
4. **Deletion at end**
5. **Deletion at given location**
6. **Deletion of given data**

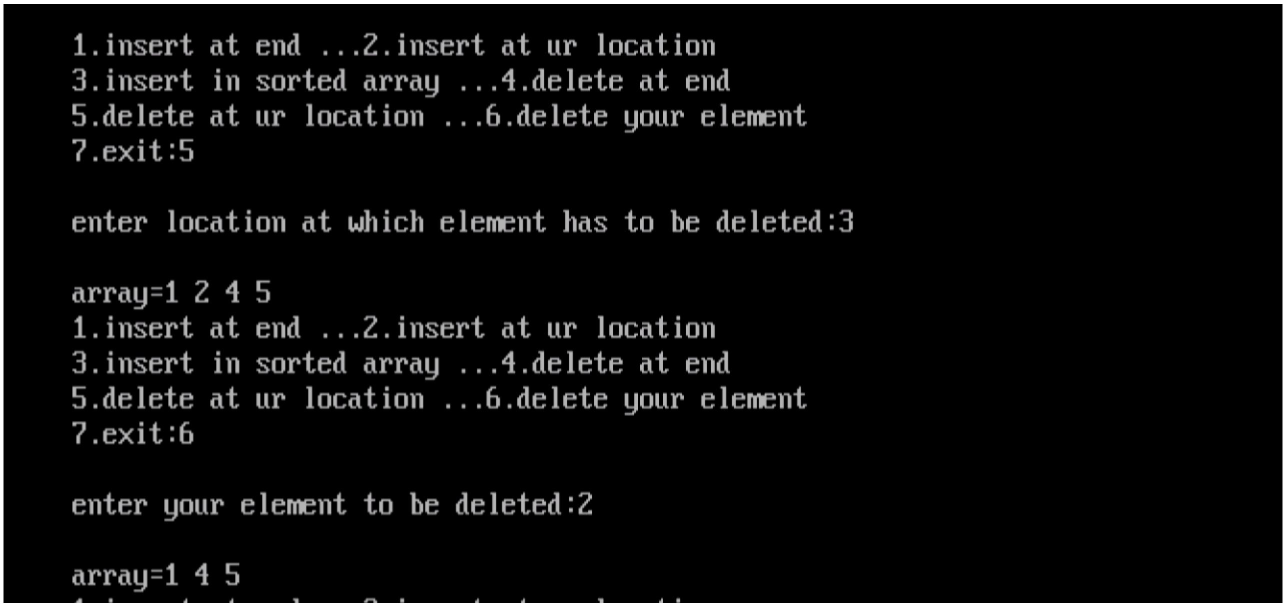
Also make a function display to display contents of array

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include<process.h>
void insert_end(int arr[], int&size,int n,int data)
{
 if(size==n)
 {
  cout<<"\noverflow";
 }
 else
 {
  arr[n]=data;
  n++;
 }
 cout<<"\narray=";
 for(int x=0;x<n;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void insertatloc(int arr[],int&n,int data,int loc)
{
 loc--;
 for(int i=n-1;i>=loc;i--)
 {
  arr[i+1]=arr[i];
 }
 arr[loc]=data;
 n++;
 cout<<"\narray=";
 for(int x=0;x<n;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void sortedarr(int arr[],int&n,int data)
{
 int i,loc=-1;
 //find location
 for(i=0;i<n;i++)
 {
  if (arr[i]<data) continue;
  else
  {
   loc=i;break;
  }
 }
 if (loc==-1)
 loc=n;
 //shift the loc
 for(i=n;i>=loc;i--)
 {
  arr[i+1]=arr[i];
 }
 arr[loc]=data;
 n++;
 cout<<"\narray=";
 for(int x=0;x<n;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void delete_end(int arr[],int &n)
{
 cout<<"\narray=";
 for(int x=0;x<n-1;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void deleteatloc(int arr[],int&n,int loc)
{
 for(int i=loc;i<n;i++)
 {
  arr[i-1]=arr[i];
 }
 n--;
 cout<<"\narray=";
 for(int x=0;x<n;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void deletebyele(int arr[],int&n,int data)
{
 int i,loc;
 //find location
 for(i=0;i<n;i++)
 {
  if(data==arr[i])
  {
   loc=i;
   break;
  }
 }
 //shift the loc
 for(i=loc;i<n;i++)
 {
  arr[i]=arr[i+1];
 }
 cout<<"\narray=";
 for(int x=0;x<n-1;x++)
 {
  cout<<arr[x]<<" ";
 }
}
void main()
{
 clrscr();
 int n,arr[10],size=10;
 cout<<"\nenter the size of the array=";
 cin>>n;
 cout<<"\nenter the array:";
 for(int a=0;a<n;a++)
 {
  cin>>arr[a];
 }
 int menu,data,loc;
 do
 {
  cout<<"\n1.insert at end ...2.insert at ur location \n3.insert in sorted array ...4.delete at
end \n5.delete at ur location ...6.delete your element \n7.exit:";
```

```
 cin>>menu;
 switch(menu)
 {
  case 1:cout<<"\n enter element to be inserted:";
           cin>>data;
           insert_end(arr,size,n,data);
           break;
  case 2:cout<<"\nenter element to be inserted:";
           cin>>data;
           cout<<"\nenter location:";
           cin>>loc;
           insertatloc(arr,n,data,loc);
           break;
  case 3:cout<<"\nenter your element to be inserted:";
           cin>>data;
           sortedarr(arr,n,data);
           break;
  case 4:delete_end(arr,n);
           break;
  case 5:cout<<"\nenter location at which element has to be deleted:";
           cin>>loc;
           deleteatloc(arr,n,loc);
           break;
  case 6:cout<<"\nenter your element to be deleted:";
           cin>>data;
           deletebyele(arr,n,data);
           break;
  case 7:exit(0);
  }
}while (menu!=7);
getch();
}
```



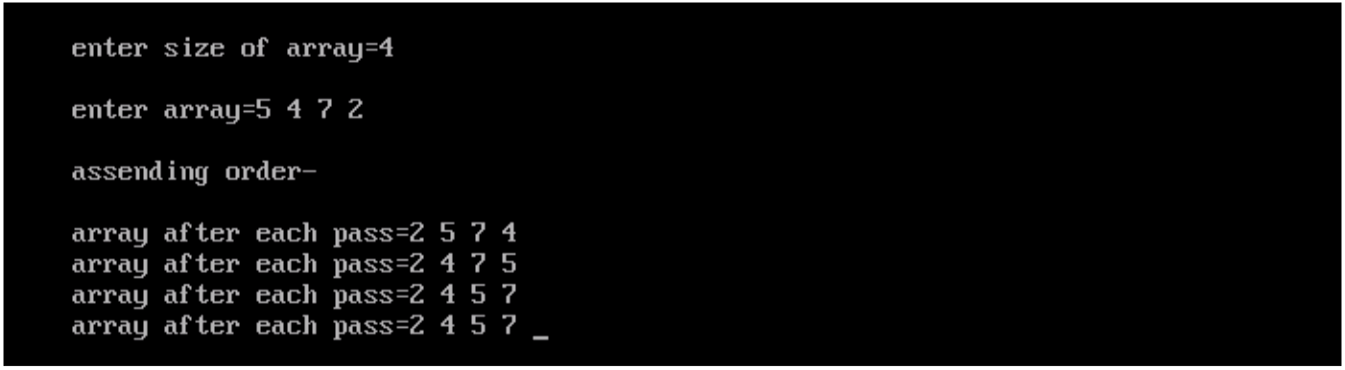
```
enter the array:1 3 5 7 9

1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:1

 enter element to be inserted:11

array=1 3 5 7 9 11
1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:2

enter element to be inserted:2

enter location:2

array=1 2 3 5 7 9
1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:3

enter your element to be inserted:4

array=1 2 3 4 5 9
1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:4

array=1 2 3 4 5
```



```
1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:5

enter location at which element has to be deleted:3

array=1 2 4 5
1.insert at end ...2.insert at ur location
3.insert in sorted array ...4.delete at end
5.delete at ur location ...6.delete your element
7.exit:6

enter your element to be deleted:2

array=1 4 5
```

19 **Make a menu based program to perform the following operations on 1D array integers**
1.Input new record and insert at the end of the array
2.Sort the array using Selection sort Technique
3.Insert new record in the sorted array
4.Display the records

```
#include<iostream.h>
#include<conio.h>
void display11(int arr[], int size)
{
 for(int i=0;i<size;i++)
 {
  cout<<arr[i]<<" ";}
 }
void selsort(int arr[],int size)
{
 int pos,temp;
 for(int i=0;i<size;i++)
 {
   for(int j=i+1;j<size;j++)
   {
    if(arr[i]>arr[j])
    {
     temp=arr[i];
     arr[i]=arr[j];
     arr[j]=temp;
  }
  cout<<"\narray after each pass=";
  display11(arr,size);
 }
}
void main()
{
 clrscr();
 int arr[10],size;
 cout<<"\nenter size of array=";
 cin>>size;
 cout<<"\nenter array=";
 for(int x=0;x<size;x++)
 {
  cin>>arr[x];
 }
 cout<<"\nassending order-"<<endl;
 selsort(arr,size);
 getch();
}
```



```
enter size of array=4

enter array=5 4 7 2

assending order-

array after each pass=2 5 7 4
array after each pass=2 4 7 5
array after each pass=2 4 5 7
array after each pass=2 4 5 7 _
```

20 Make a menu based program to perform the following operations on 1D array of integers

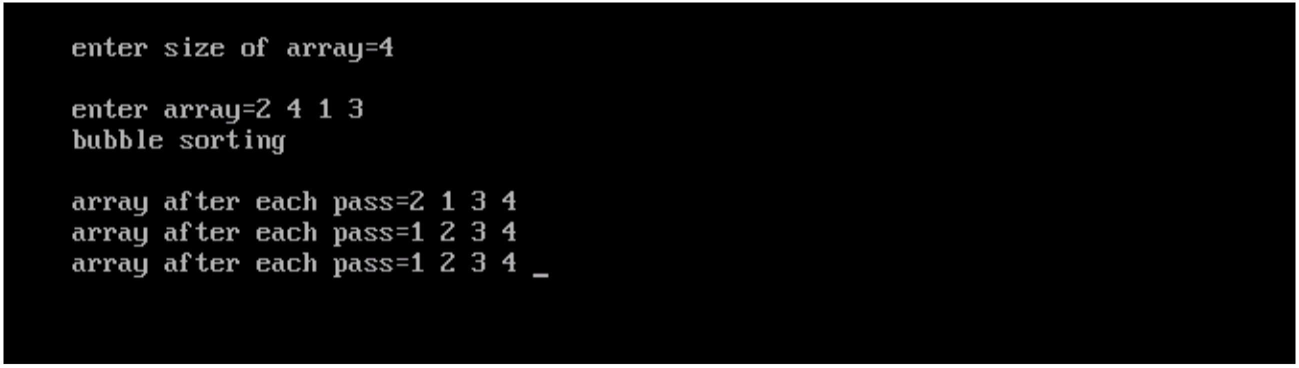
1. Input new record and insert at the end of the array

2. Sort the array using Bubble sort technique

3. Delete one number from end in the array.

4. Display the elements

```
#include<iostream.h>
#include<conio.h>
void display(int arr[], int size)
{
 for(int i=0;i<size;i++)
 {
  cout<<arr[i]<<" ";
 }
}
void bubsort(int arr[],int size)
{
 int pos,temp;
 for(int i=0;i<size-1;i++)
 {
  for(int j=0;j<size-i-1;j++)
  {
   if(arr[j]>arr[j+1])
   {
    temp=arr[j];
    arr[j]=arr[j+1];
    arr[j+1]=temp;
   }
  }
 cout<<"\narray after each pass=";
 display(arr,size);
 }
}
void main()
{
 clrscr();
 int arr[10],size;
 cout<<"\nenter size of array=";
 cin>>size;
 cout<<"\nenter array=";
 for(int x=0;x<size;x++)
 {
  cin>>arr[x];
 }
 cout<<"bubble sorting"<<endl;
 bubsort(arr,size);
 getch();
}
```



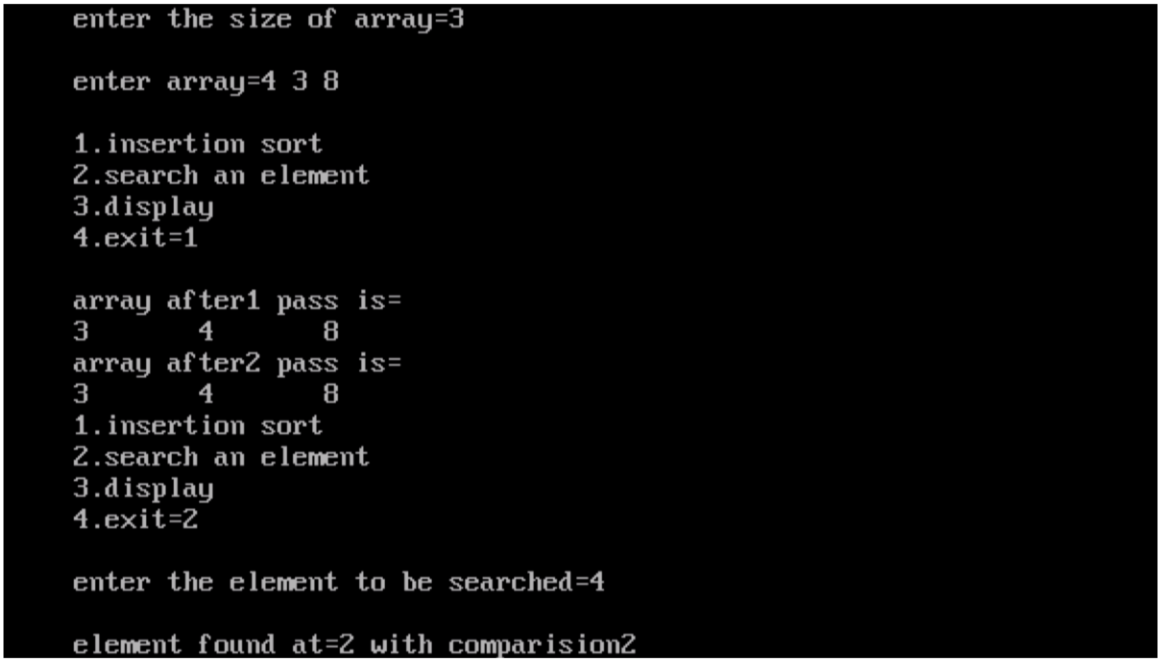
```
enter size of array=4

enter array=2 4 1 3
bubble sorting

array after each pass=2 1 3 4
array after each pass=1 2 3 4
array after each pass=1 2 3 4 _
```

21 Make a menu based program to perform the following operations on 1D array of integers

1. **Input new record and insert at the end of the array**
2. **Sort the array using Insertion Sort technique**
3. **Search a particular number using Linear Search method**
4. **Display the records.**

```
#include<conio.h>
#include<iostream.h>
#include<process.h>
void display(int arr[],int size)
{
 cout<<"\narray-";
 for(int x=0;x<size;x++)
 {
  cout<<arr[x];
 }
}
void insertion(int a[],int n)
{
 int i,pass,curr,pos;
 for(pass=1;pass<n;pass++)
 {
  curr=a[pass];
  pos=-1;
  for(i=0;i<=pass;i++)
  {
   if(a[i]<=curr)
   pos++;
  }
  if(pos!=pass)
  {
   for(i=pass-1;i>=pos;i--)
   a[i+1]= a[i];
   a[pos]=curr;
  }
  cout<<"\narray after"<<pass<<" pass is=\n";
  for(int k=0;k<n;k++)
  cout<<a[k]<<"\t";
 }
}
```

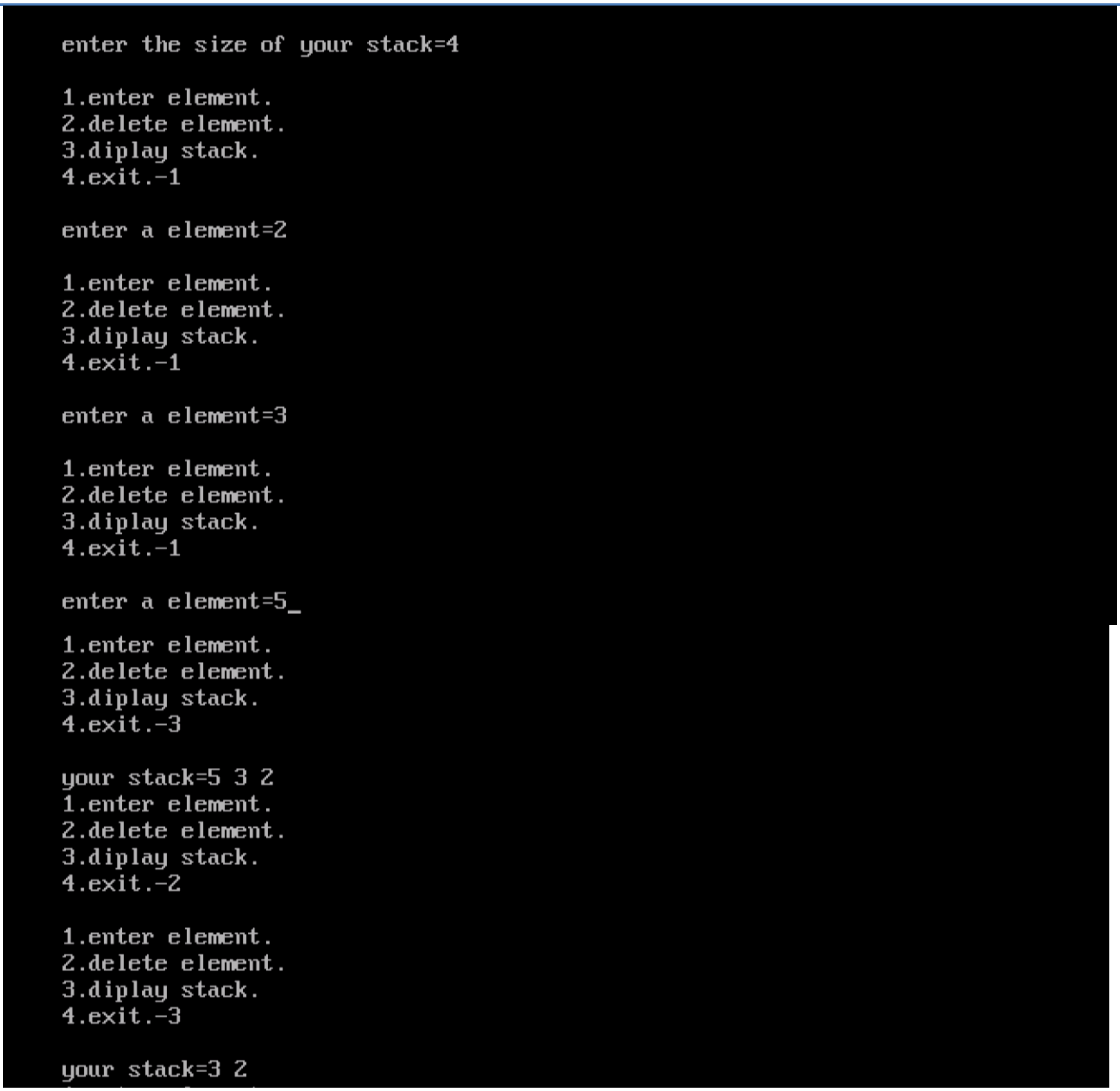
```
void lsearch(int arr[],int size,int ele)
{
 int flag=0,comp=1;
 for (int x=0;x<size;x++)
 {
  if(ele==arr[x])
  {
   cout<<"\nelement found at="<<x+1<<" with comparision"<<comp;
   flag++;break;
  }
  else comp++;
 }
 if(flag==0)
 cout<<"\nelement not found";
}
void main()
{
 clrscr();
 int arr[20],size,ele;
 cout<<"\nenter the size of array=";
 cin>>size;
 cout<<"\nenter array=";
 for(int x=0;x<size;x++)
 {
  cin>>arr[x];
 }
 int menu;
 do
 {
  cout<<"\n1.insertion sort\n2.search an element\n3.display\n4.exit=";
  cin>>menu;
  switch(menu)
  {
  case 1:insertion(arr,size); break;
  case 2:cout<<"\nenter the element to be searched=";
         cin>>ele;
         lsearch(arr,size,ele);break;
  case 3:display(arr,size);break;
  case 4: exit(0);
  }
 }while(menu!=3);
getch(); }
```



```
enter the size of array=3

enter array=4 3 8

1.insertion sort
2.search an element
3.display
4.exit=1

array after1 pass is=
3       4       8
array after2 pass is=
3       4       8
1.insertion sort
2.search an element
3.display
4.exit=2

enter the element to be searched=4

element found at=2 with comparision2
```

22 Make a menu based program to implement stack using array 1.Push 2. Pop 3. Display

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
int a[10],size;
int top=-1;
void push()
{
 if(top==size-1)
 {
  cout<<"\nempty full.";
 }
 else
 {
  int ele;
  cout<<"\nenter a element=";
  cin>>ele;
  top++;
  a[top]=ele;
 }
}
void pop()
{
 if(top==-1)
 {
  cout<<"\nstack is empty.";
 }
 else
  top--;
}
void display()
{
 int x=top;
 cout<<"\nyour stack=";
 while(x>=0)
 {
  cout<<a[x]<<" ";
  x--;
 }
}
void main()
{
clrscr();
int menu;
cout<<"\nenter the size of your stack=";cin>>size;
do
{
 cout<<"\n1.enter element.\n2.delete element.\n3.diplay stack.\n4.exit.-";
 cin>>menu;
 switch(menu)
 {
  case 1:push();break;
  case 2:pop();break;
  case 3:display();break;
  case 4:exit(0);break;
 }
}while(menu!=4);
getch();
}
```

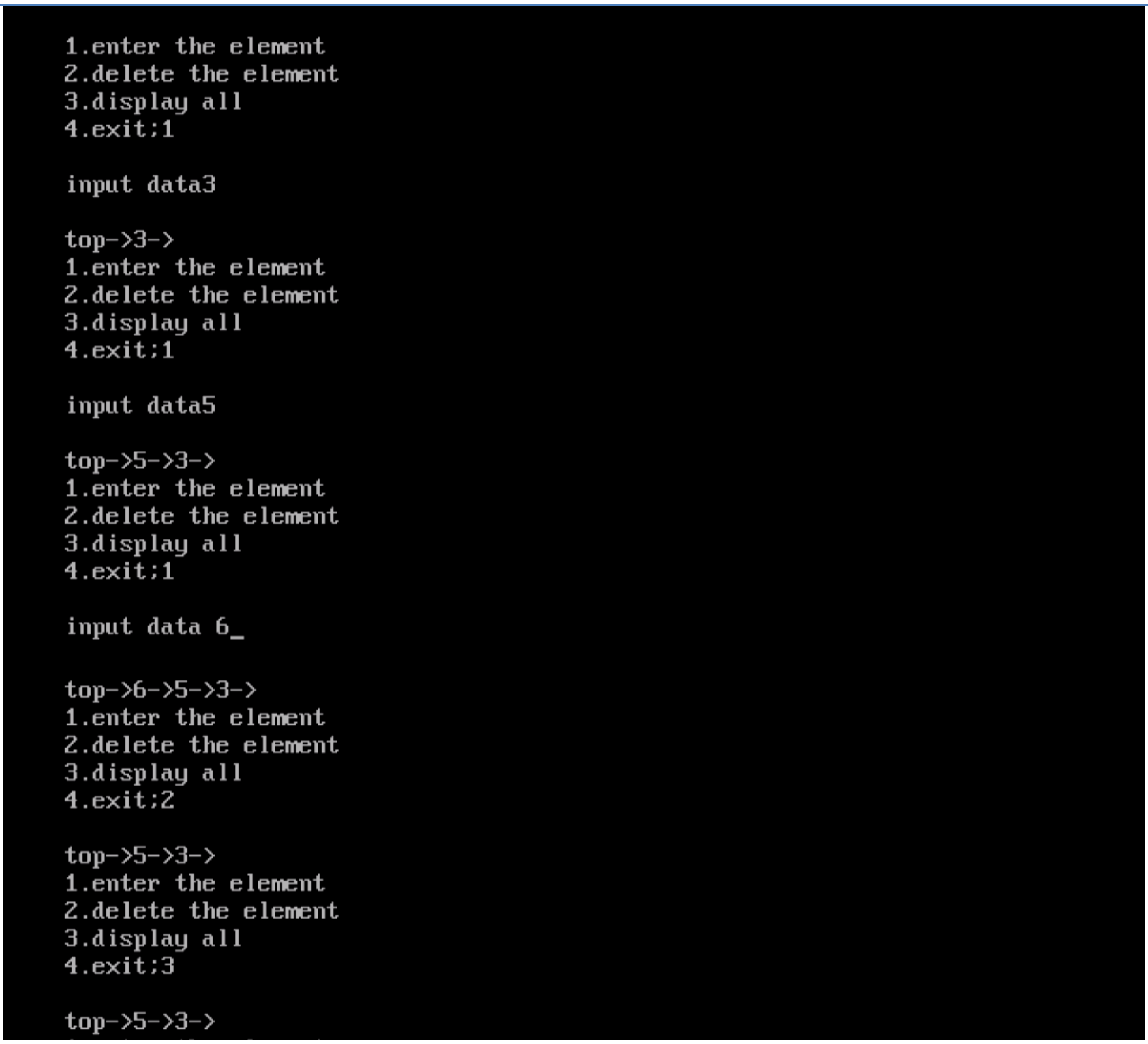


```
enter the size of your stack=4

1.enter element.
2.delete element.
3.diplay stack.
4.exit.-1

enter a element=2

1.enter element.
2.delete element.
3.diplay stack.
4.exit.-1

enter a element=3

1.enter element.
2.delete element.
3.diplay stack.
4.exit.-1

enter a element=5_
1.enter element.
2.delete element.
3.diplay stack.
4.exit.-3

your stack=5 3 2
1.enter element.
2.delete element.
3.diplay stack.
4.exit.-2

1.enter element.
2.delete element.
3.diplay stack.
4.exit.-3

your stack=3 2
```

23 Make a menu based program to implement Stack using Linked lists 1. Push 2.Pop 3. Display

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
struct node
{
 int info;
 node *next;
};
```

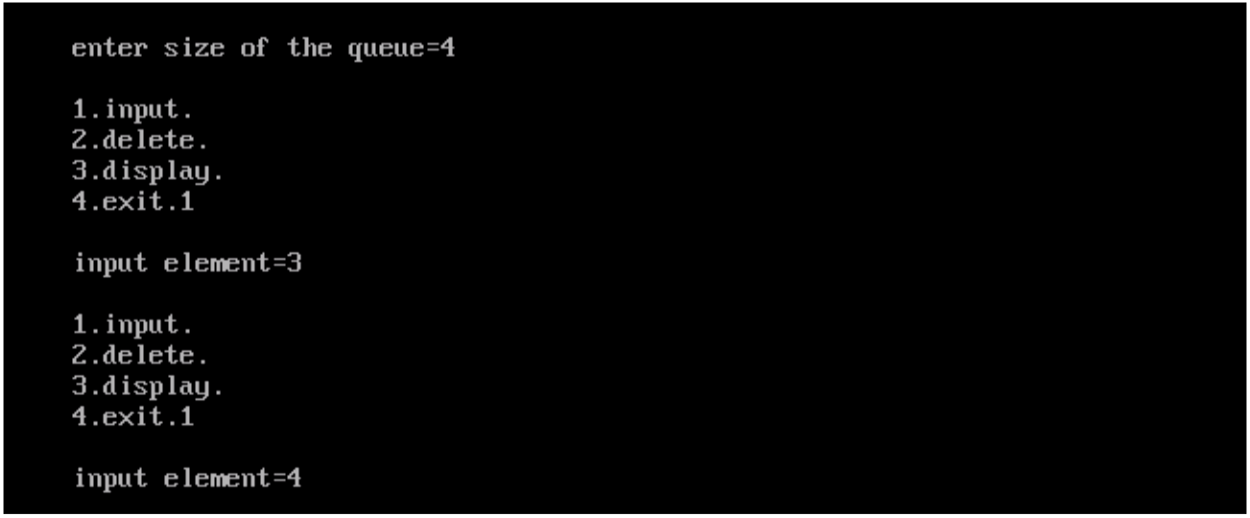
```
class stack
{
 private:node *top;
 public:
                        stack()
                        {
                         top=NULL;
                        }
                        void push();
                        void pop();
                        void display();
};
void stack::push()
{
 int ele;
 cout<<"\ninput data";
 cin>>ele;
 node*ptr=new node;
 ptr->info=ele;
 ptr->next=NULL;
 if(top==NULL)
 {
  top=ptr;
 }
 else
 {
  ptr->next=top;
  top=ptr;
 }
 display();
}
void stack::display()
{
 node *temp=top;
 if(temp==NULL)
  cout<<"\nstack is empty;";
 else
 {
  cout<<"\ntop->";
  while(temp!=NULL)
  {
   cout<<temp->info<<"->";
   temp=temp->next;
  }
 }
}
void stack::pop()
{
 if(top==NULL)
  cout<<"\nstack is emptyt";
 else
 {
  node *temp=top;
  top=top->next;
  delete temp;
 }
 display();
}
void main()
{
 clrscr();
 stack st;
 int menu;
 do
 {
  cout<<"\n1.enter the element \n2.delete the element \n3.display all \n4.exit;";
  cin>>menu;
  switch(menu)
  {
   case 1:st.push();break;
   case 2:st.pop();break;
   case 3:st.display();break;
   case 4:exit(0);break;
  }
 }while(menu!=4);
 getch();
}
```



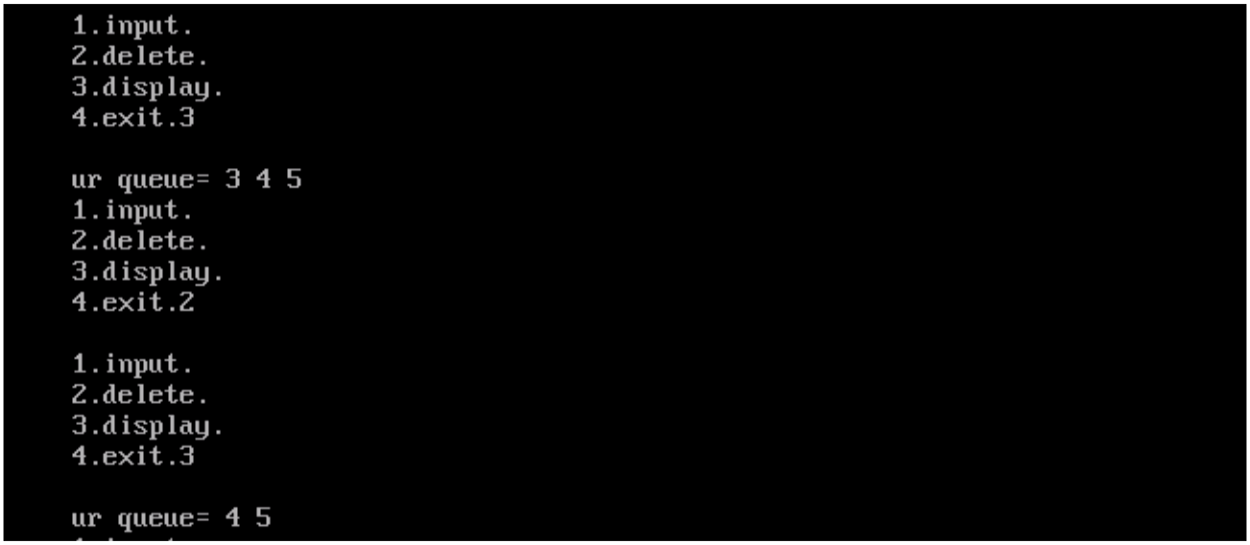
```
1.enter the element
2.delete the element
3.display all
4.exit;1

input data3

top->3->
1.enter the element
2.delete the element
3.display all
4.exit;1

input data5

top->5->3->
1.enter the element
2.delete the element
3.display all
4.exit;1

input data 6_

top->6->5->3->
1.enter the element
2.delete the element
3.display all
4.exit;2

top->5->3->
1.enter the element
2.delete the element
3.display all
4.exit;3

top->5->3->
```

24 Make a menu based program to implement Queue using array 1. Insert 2. Remove 3. Display

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
int a[20],size;
int front=-1;
int rear=-1;
void enqueue()
{
 int ele;
 if(rear==size-1)
 cout<<"\nqueue is full.";
 else
 {
  cout<<"\ninput element=";
  cin>>ele;
  rear++;
  a[rear]=ele;
 }
 if(front==-1)
  front=0;
}
void dequeue()
{
 if(front==-1)
 cout<<"\nstack is empty.";
 else
 front++;
 if(front==rear+1)
 front=-1;
}
void display()
{
 int x=front;
 cout<<"\nur queue= ";
 while(x<=rear)
 {
  cout<<a[x]<<" ";
  x++;
 }
}
void main()
{
 clrscr();
 int menu;
 cout<<"\nenter size of the queue=";
 cin>>size;
 do
 {
  cout<<"\n1.input.\n2.delete.\n3.display.\n4.exit.";
  cin>>menu;
  switch(menu)
  {
   case 1:enqueue();break;
   case 2:dequeue();break;
   case 3:display();break;
   case 4:exit(0);
  }
 }while(menu!=4);
 getch();
}
```



```
enter size of the queue=4

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

input element=3

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

input element=4
```



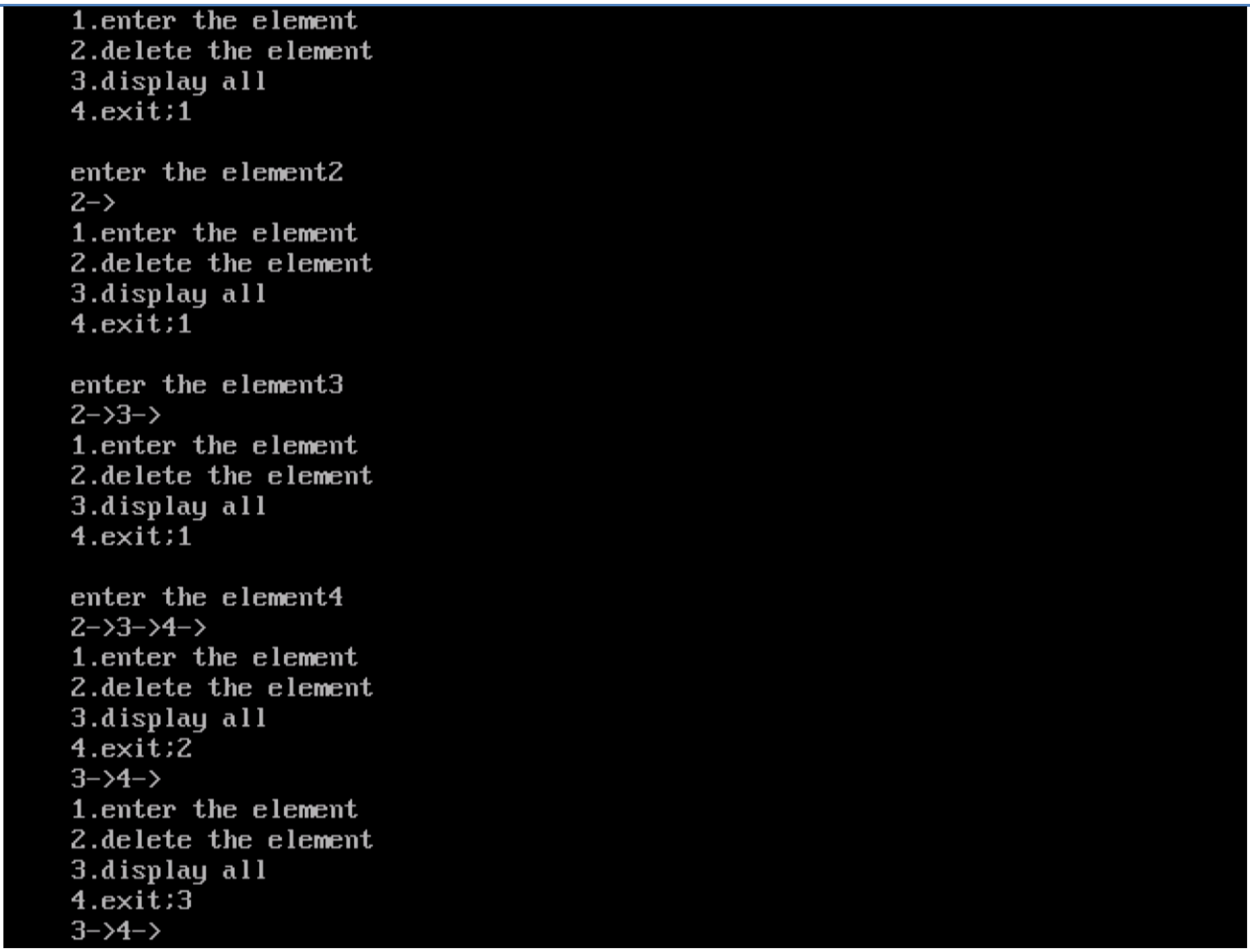
```
1.input.
2.delete.
3.display.
4.exit.3

ur queue= 3 4 5
1.input.
2.delete.
3.display.
4.exit.2

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

ur queue= 4 5
```

25 Make a menu based program to implement Queue using Linked Lists 1. Insert 2. Remove 3. Display

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
struct node
{
```

```
int info; node *next;
};
class queue
{
private: node *rear, *front;
public:queue()
       {
       rear=front=NULL;
       }
       void insert(int);
       void remove();
       void display();
};
void queue:: insert(int ele)
{
 node *ptr=new node;
 ptr->info=ele;
 ptr->next=NULL;
 if(front==NULL)
 {
  front=ptr;rear=ptr;
 }
 else
 {
  rear->next=ptr; rear=ptr;
 }
 display();
}
void queue::remove()
{
 if (front==NULL)
 cout<<"\nqueue is empty";
 else
 {
  node *temp=front;
  front=front->next;
  if(front==NULL)
  rear=NULL;
  delete temp;
 }
 display();
}
```

```
void queue::display()
{
 node*temp=front;
 if(temp==NULL)cout<<"\nqueue is empty!!";
 else
 {
  while(temp!=NULL)
  {
   cout<<temp->info<<"->";
   temp=temp->next;
  }
 }
}
void main()
{
 clrscr();
 queue q;
 int menu,ele;
 do
 {
  cout<<"\n1.enter the element \n2.delete the element \n3.display all \n4.exit;";
  cin>>menu;
  switch(menu)
  {
   case 1:cout<<"\nenter the element";
                          cin>>ele;
                          q.insert(ele);
                          break;
   case 2:q.remove();break;
   case 3:q.display();break;
   case 4:exit(0);break;
  }
 }while(menu!=4);
 getch(); }
```



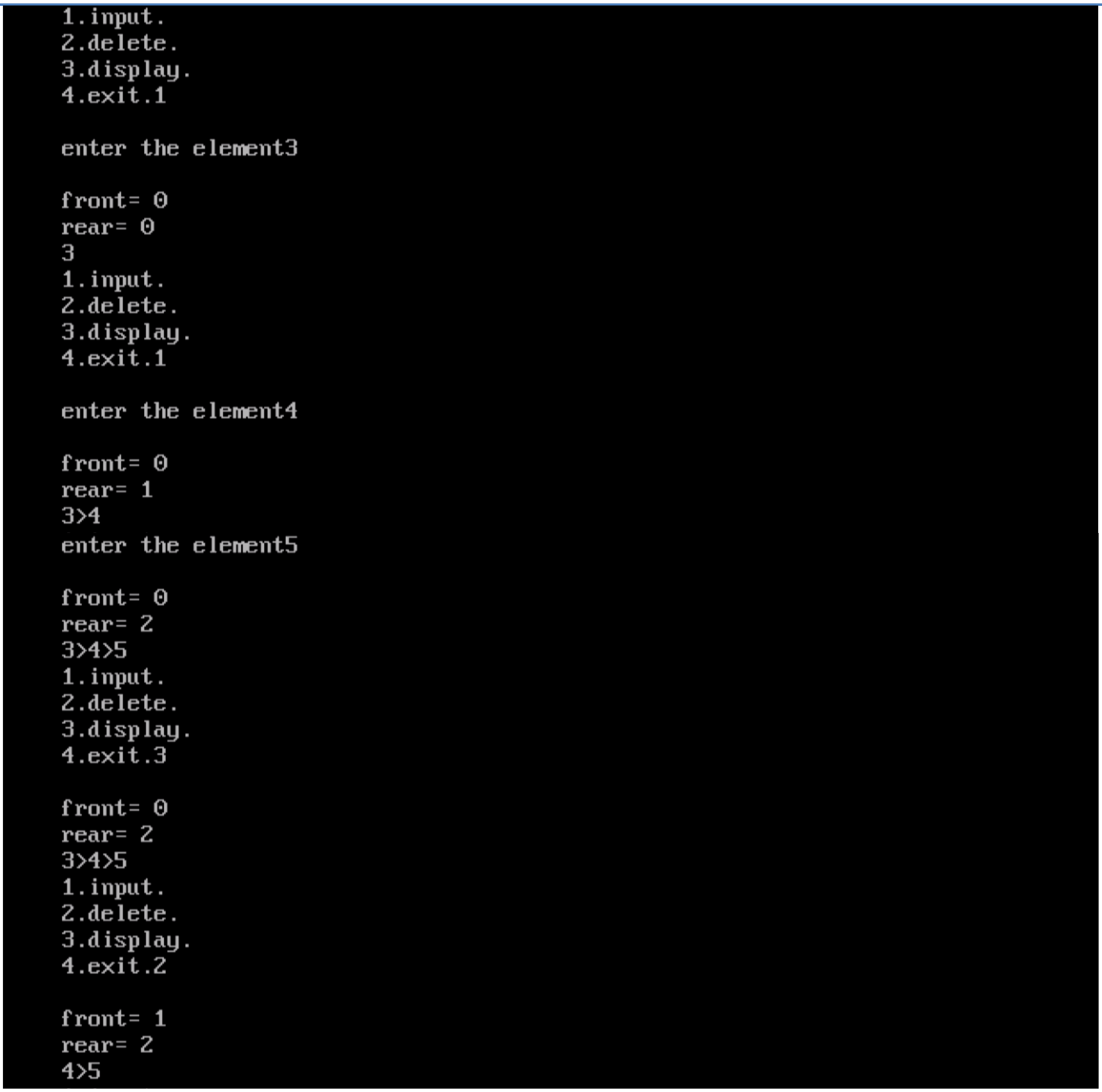
```
1.enter the element
2.delete the element
3.display all
4.exit:1

enter the element2
2->
1.enter the element
2.delete the element
3.display all
4.exit:1

enter the element3
2->3->
1.enter the element
2.delete the element
3.display all
4.exit:1

enter the element4
2->3->4->
1.enter the element
2.delete the element
3.display all
4.exit:2
3->4->
1.enter the element
2.delete the element
3.display all
4.exit:3
3->4->
```

26 Make a menu based program to implement Circular Queue using array 1. Insert 2. Remove 3. Display

```
#include<iostream.h>
#include<conio.h>
#include<process.h>
class cqueue
{
private:int data[5];
        int rear,front;
public: cqueue()
        {
         rear=front=-1;
        }
        void cinsert(int ele)
        {
         if(rear==4&&front==0||front==rear+1)
         {
```

```
  cout<<"queue is full/";
  return;
 }
 if(rear==-1)
 {
  rear++;
  front++;
 }
 else if(rear==4&&front!=0)
 rear=0;
 else
 rear++;
 data[rear]=ele;
 cdisplay();
}
void cremove()
{
 if(front==-1)
 cout<<"\nqueue is empty";
 else
 {
  if(front==rear)
  {
   front=-1;
   rear=-1;
  }
  else if(front==4)
  front=0;
  else
  front++;
 }
 cdisplay();
}
void cdisplay()
{
 int temp=front;
 if(temp==-1)
 cout<<"\nqueue is empty";
 else
 {
  cout<<"\nfront= "<<front<<"\nrear= "<<rear<<"\n";
  while(temp!=rear)
```

```
            {
             cout<<data[temp]<<">";
             temp=(temp+1)%5;
            }
            cout<<data[temp];
           }
          }
};
void main()
{
 clrscr();
 int menu,ele;
 cqueue c;
 do
 {
  cout<<"\n1.input.\n2.delete.\n3.display.\n4.exit.";
  cin>>menu;
  switch(menu)
  {
   case 1:cout<<"\nenter the element";cin>>ele;
            c.cinsert(ele);break;
   case 2:c.cremove();break;
   case 3:c.cdisplay();break;
   case 4:exit(0);
  }
 }while(menu!=4);
 getch();
}
```



```
1.input.
2.delete.
3.display.
4.exit.1

enter the element3

front= 0
rear= 0
3
1.input.
2.delete.
3.display.
4.exit.1

enter the element4

front= 0
rear= 1
3>4
enter the element5

front= 0
rear= 2
3>4>5
1.input.
2.delete.
3.display.
4.exit.3

front= 0
rear= 2
3>4>5
1.input.
2.delete.
3.display.
4.exit.2

front= 1
rear= 2
4>5
```