**Question 1**

**An electricity board charges the following rates to user.**

**For the first 100 units → 60p per unit.**

**For the next 200 units→80p per unit.**

**Beyond 300 units→90p per unit.**

**1).All users are charged a minimum of Rs. 50; if the total amount is more than 300 then an additional surcharges of 15% is added. Write a program to accept name of user consumed and print charges with their rates**.

**Solution**

#include<iostream.h>

#include<conio.h>

const int minimum=50;

void main()

{

float total,unit,additional=0;

char name[20];

clrscr();

cout<<"Enter Name :- ";

cin>>name;

cout<<"Enter Unit :- ";

cin>>unit;

if(unit<=100)

total=(unit\*0.6)+minimum;

else if(unit<=300)

total=(unit\*0.6)+((unit-100)\*0.8)+minimum;

else

total=(unit\*0.6)+((unit-100)\*0.8)+((unit-300)\*0.9)+minimum;

if(total>300)

{

additional=total\*0.15;

total+=additional;

}

cout<<"Name :- "<<name<<endl;

cout<<"Units Consumed :- "<<unit<<endl;

cout<<"Additional Charges :- "<<additional<<endl;

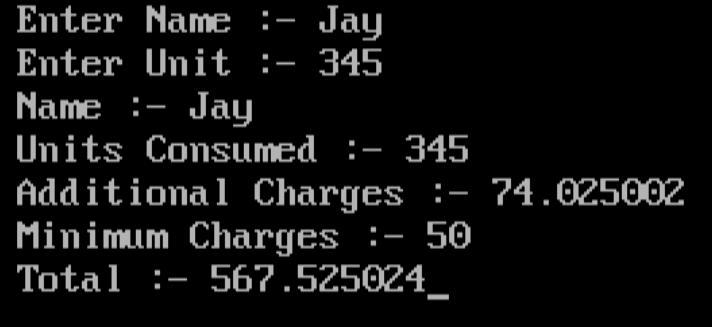
cout<<"Minimum Charges :- "<<minimum<<endl;

cout<<"Total :- "<<total;

getch();

}

**Output**



**Question 2**

**Define a class to represent a bank account. Include the following members:**

**a. Name of the depositor b. Account number**

**c. Type of Account d. Balance amount in the Account Member Functions:**

1. **To assign initial values.**
2. **To deposit an amount.**
3. **To withdraw an amount after checking the balance.**
4. **To display name and balance.**

**Write main program and handle accounts of 5 customers.**

**Solution**

#include<iostream.h>

#include<conio.h>

class bank

{

protected:

char name[20],type[10];

float amount;

public:

int acno;

void getdata()

{

cout<<"\n\nEnter Name :- ";

cin>>name;

cout<<"Enter Account Number :- ";

cin>>acno;

cout<<"Enter Type Of Account :- ";

cin>>type;

cout<<"Enter Deposited Amount To Open Account :-";

cin>>amount;

cout<<"Your Account Opened Successfully..."<<endl;

}

void deposit()

{

float d;

cout<<"\n\nEnter Amount to Deposit :- ";

cin>>d;

amount+=d;

cout<<"Amount Successfully Deposited To your Account..."<<endl;

}

void withdraw()

{

float w;

cout<<"\n\nEnter Amount To Withdraw :- ";

cin>>w;

if(w>amount)

{

cout<<"You have insufficient Balance..."<<endl;

return;

}

amount-=w;

cout<<"Amount Successfully Withdrawn..."<<endl;

}

void display()

{

cout<<"\n\nName :- "<<name<<endl;

cout<<"Balance :- "<<amount<<endl;

}

};

void main()

{

int ch,i,acc,j=0,choice;

bank b[5];

clrscr();

do

{

cout<<"\n1.New Account\n2.Existing Account\n3.Exit\n";

cin>>ch;

switch(ch)

{

case 1:

b[j].getdata();

j++;

break;

case 2:

cout<<"\nEnter Account Number :- ";

cin>>acc;

for(i=0;i<j;i++)

{

if(b[i].acno==acc)

{

do

{

cout<<"\n1.Deposit\n2.Withdraw\n3.Display\n4.Exit\n";

cin>>choice;

switch(choice)

{

case 1:

b[i].deposit();

break;

case 2:

b[i].withdraw();

break;

case 3:

b[i].display();

break;

case 4:

cout<<"Exitting..."<<endl;

break;

default:

cout<<"Enter Corrct Choice...";

}

}while(choice!=4);

break;

}

}

if(i==j)

cout<<"Account Not Found..."<<endl;

break;

case 3:

cout<<"Exitting..."<<endl;

break;

default:

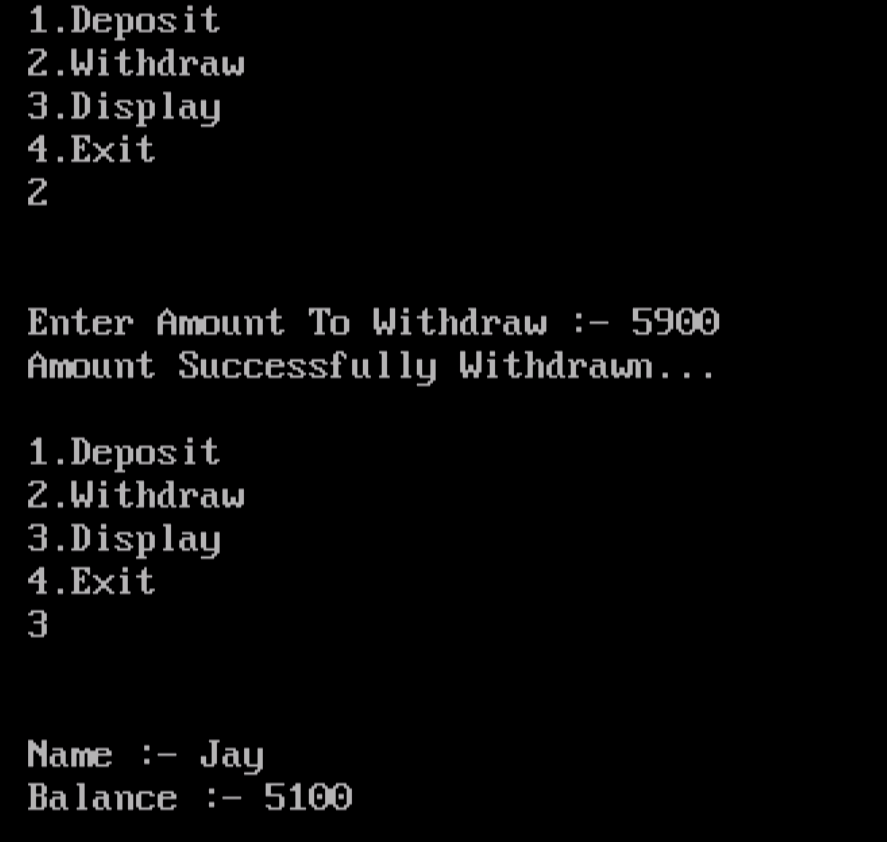
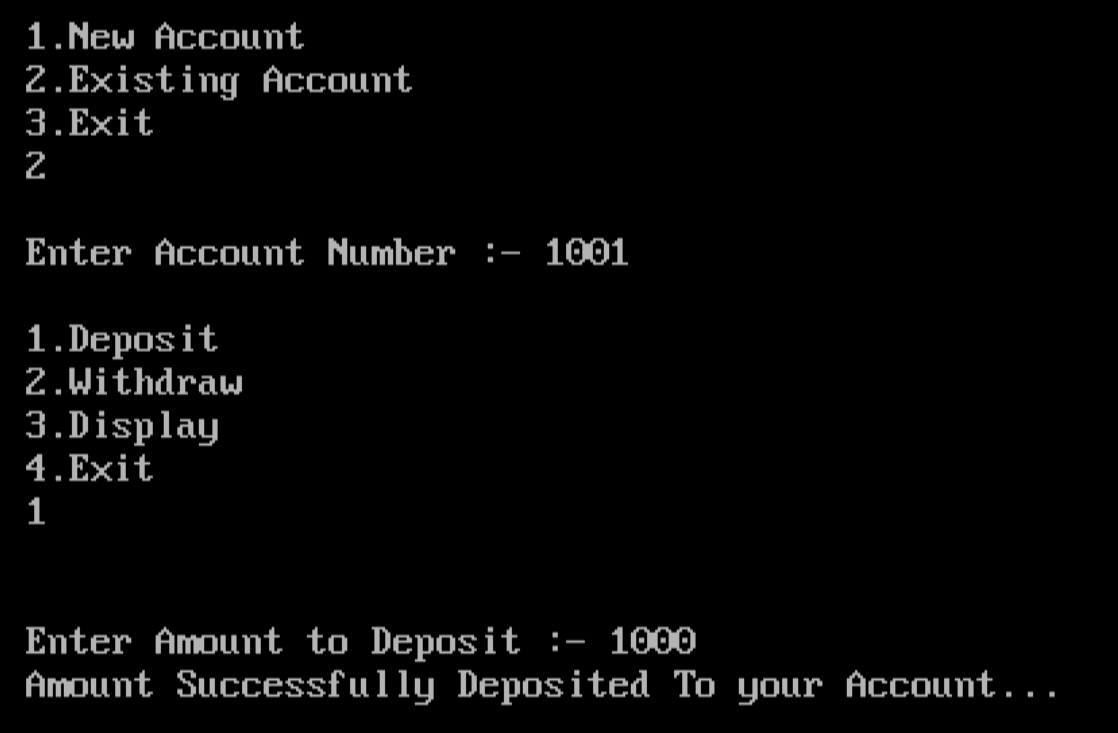
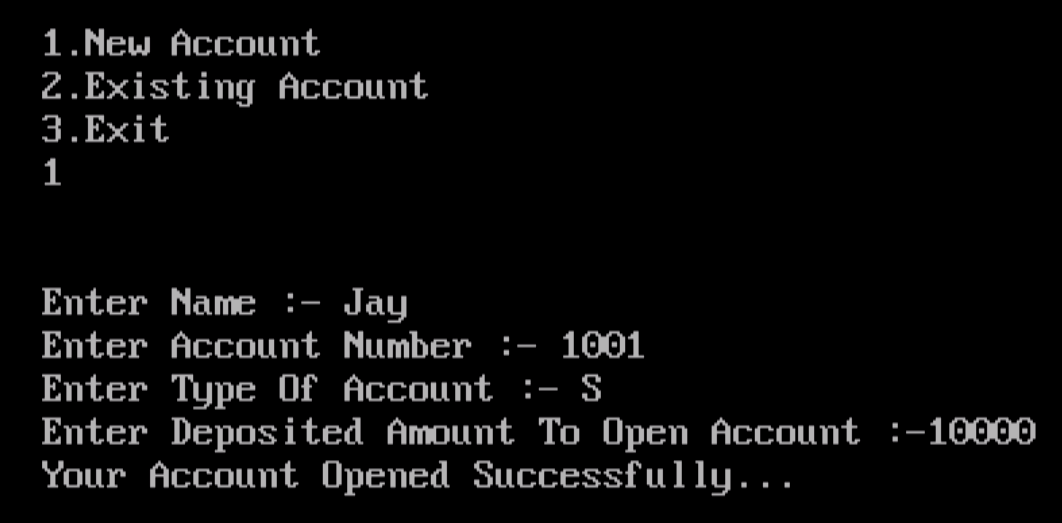
cout<<"Enter Correct Choice...";

}

}while(ch!=3);

}

**Output**



**Question 3**

**Program to create a class person having members name and age. Derive a class student having member percentage. Derive another class teacher having member salary. Write necessary member function to initialize, read and write data. Also write the main function.**

**Solution**

#include<iostream.h>

#include<conio.h>

class Person

{

protected:

char name[20];

int age;

public:

void getdata()

{

cout<<"Enter Name :- ";

cin>>name;

cout<<"Enter Age :- ";

cin>>age;

}

void display()

{

cout<<"Name :- "<<name<<endl;

cout<<"Age :- "<<age<<endl;

}

};

class Student:public Person

{

protected:

float per;

public:

void getstu()

{

getdata();

cout<<"Enter Percentage :- ";

cin>>per;

}

void disstu()

{

display();

cout<<"Percentage :- "<<per<<endl;

}

};

class Teacher:public Person

{

protected:

float salary;

public:

void gettea()

{

getdata();

cout<<"Enter Salary :- ";

cin>>salary;

}

void distea()

{

display();

cout<<"Salary :- "<<salary<<endl;

}

};

void main()

{

int ch,i,sn,tn;

Student s[10];

Teacher t[10];

do

{

cout<<"\n1.Enter Student Details\n2.Display Student\n3.Enter Teacher Details\n4.Display Teacher Details\n5.Exit";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Number Of Students :- ";

cin>>sn;

for(i=0;i<sn;i++)

{

cout<<"Enter Details Of Student "<<i+1<<" :-\n\n";

s[i].getstu();

}

break;

case 2:

for(i=0;i<sn;i++)

{

cout<<"Details Of Student "<<i+1<<" :-\n\n";

s[i].disstu();

}

break;

case 3:

cout<<"Enter Number Of Teachers :- ";

cin>>tn;

for(i=0;i<tn;i++)

{

cout<<"Enter Details Of Teacher "<<i+1<<" :-\n\n";

t[i].gettea();

}

break;

case 4:

for(i=0;i<tn;i++)

{

cout<<"Details Of Teacher "<<i+1<<" :-\n\n";

t[i].distea();

}

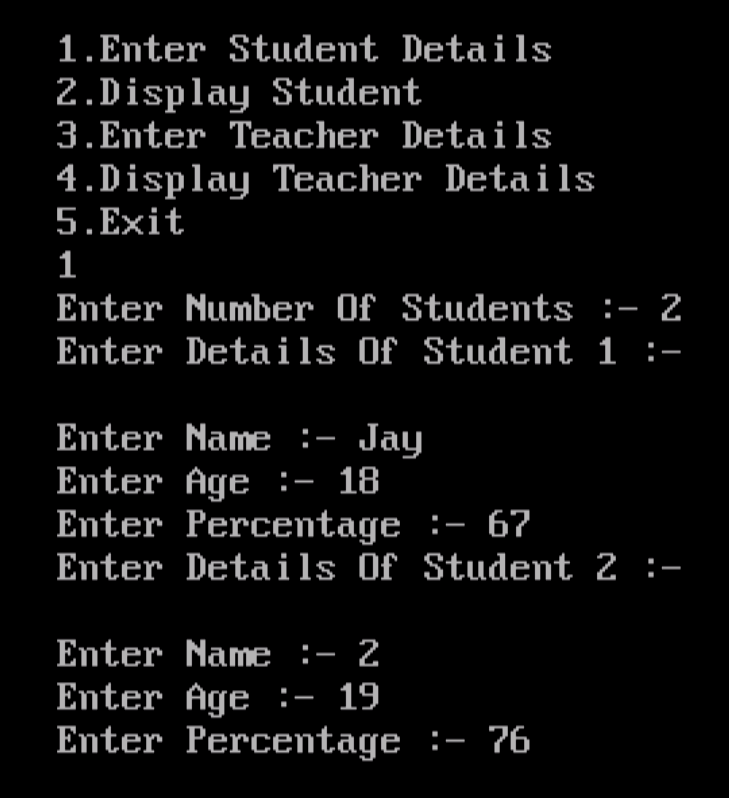
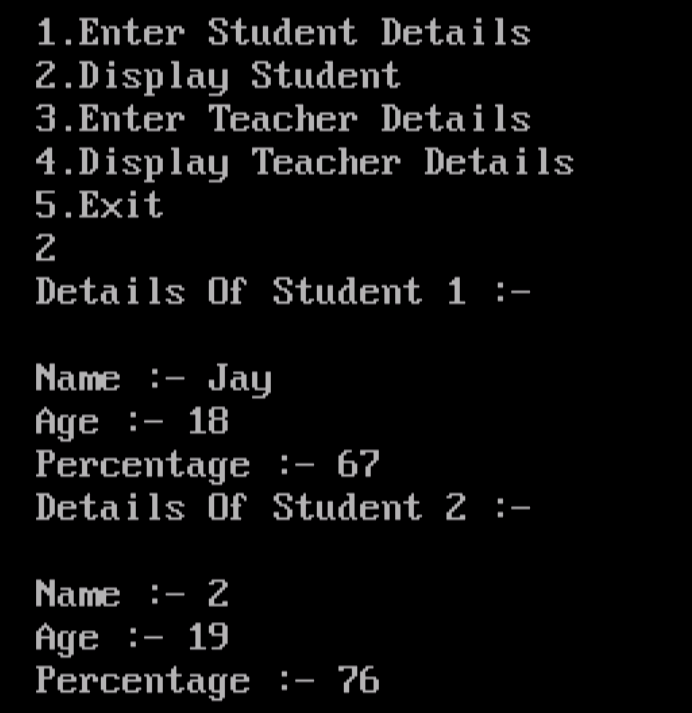
break;

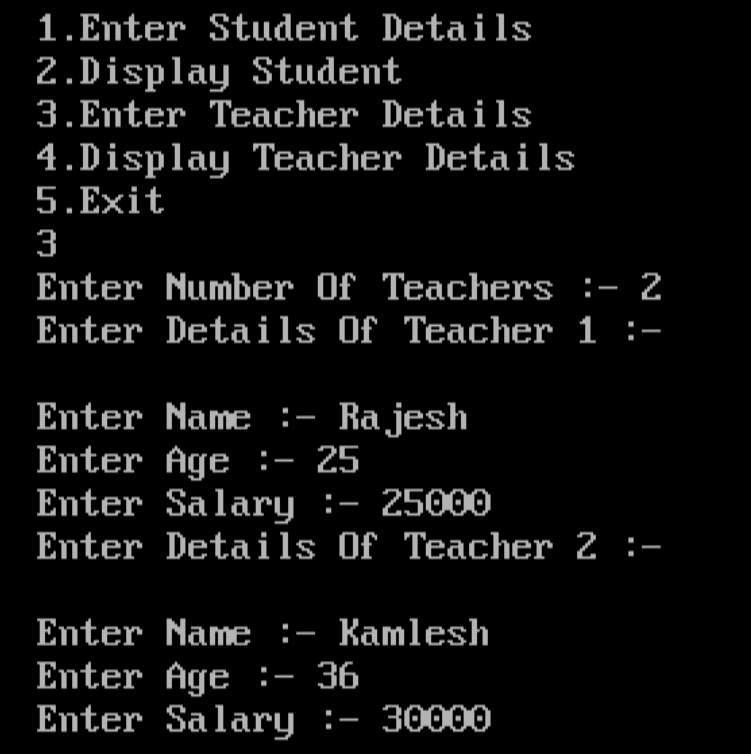
}

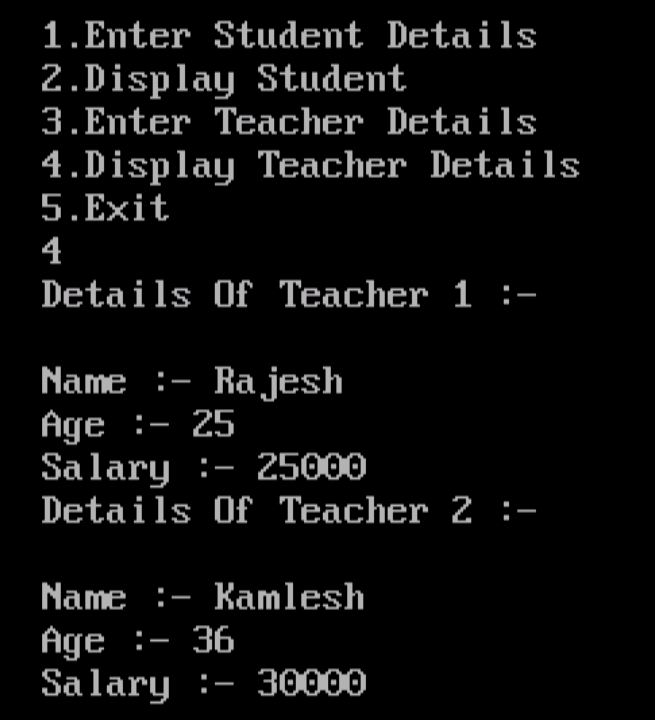
}while(ch!=5);

}

**Output**







**Question 4**

**Program to create a class name student having date member name, no & three marks. Write a member function to input name, roll no & marks & calculate percentage.**

**Solution**

#include<iostream.h>

#include<conio.h>

class Student

{

protected:

char name[20];

int no,m1,m2,m3;

float per;

public:

void getdata()

{

cout<<"Enter Roll No :- ";

cin>>no;

cout<<"Enter Name :- ";

cin>>name;

cout<<"Enter Marks1 :- ";

cin>>m1;

cout<<"Enter Marks2 :- ";

cin>>m2;

cout<<"Enter Marks3 :- ";

cin>>m3;

}

void percentage()

{

getdata();

per=(m1+m2+m3)/3.0;

cout<<"Percentage :- "<<per;

}

};

void main()

{

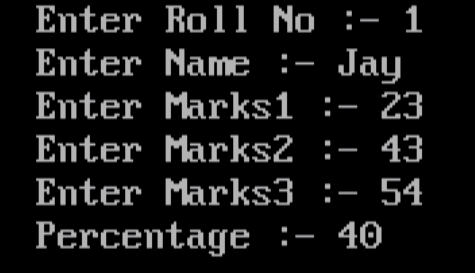
Student s;

clrscr();

s.percentage();getch();

}

**Output**



**Question 5**

**Create a class called "Vehicle" which contains data members registration number and fuel type Make getdata() function to input data value. Create class "two-Wheeler" from vehicle which contains data member’s distance and mileage Make getdata() function to input data. Use overloading techniques for getdata() function and display the information with fuel used.**

**Solution**

#include<iostream.h>

#include<conio.h>

class Vehicle

{

protected:

char rn[20],ftype[10];

public:

virtual void getdata()

{

cout<<"Enter Registration Number :- ";

cin>>rn;

cout<<"Enter Fuel Type :- ";

cin>>ftype;

}

virtual void display()

{

cout<<"\n\nRegistration Number :- "<<rn<<endl;

cout<<"Fuel Type :- "<<ftype<<endl;

}

};

class Two\_wheeler:public Vehicle

{

protected:

float distance,mileage;

public:

void getdata()

{

cout<<"Enter Distance Of Your Vehicle Travelled(km) :- ";

cin>>distance;

cout<<"Enter Mileage Of Your Vehicle(km/l) :- ";

cin>>mileage;

}

void display()

{

cout<<"Distance Travelled :- "<<distance<<endl;

cout<<"Mileage :- "<<mileage<<endl;

cout<<"Fuel Used :- "<<distance\*mileage<<endl;

}

};

void main()

{

Vehicle v;

Two\_wheeler t;

Vehicle \*v1=&v,\*t1=&t;

clrscr();

v1->getdata();

t1->getdata();

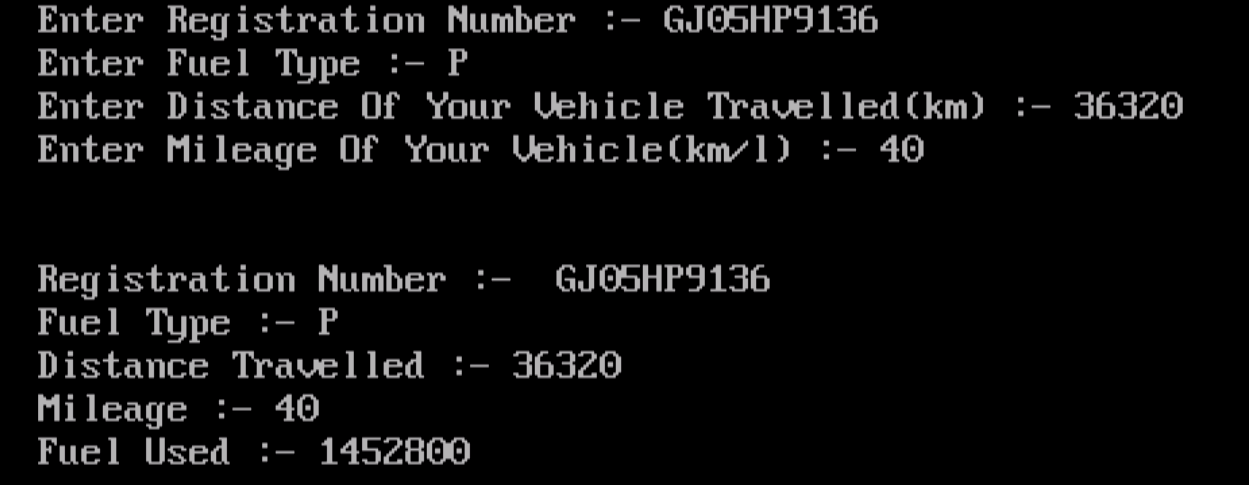
v1->display();

t1->display();

getch();

}

**Output**

****

**Question 6**

**Write a program that consist of two classes Time12 and Time24. The first one maintains time on 12 hour basis, whereas the other one maintains it on 24-hour basis.**

**Solution**

#include<iostream.h>

#include<conio.h>

class Time12

{

protected:

int h,m,s;

char mer[2];

public:

void gettime()

{

cout<<"Enter Hour :- ";

cin>>h;

cout<<"Enter Minute :- ";

cin>>m;

cout<<"Enter Second :- ";

cin>>s;

cout<<"Enter Meridian(AM/PM) :- ";

cin>>mer;

}

void display()

{

gettime();

cout<<"Time :-\n";

cout<<h<<":"<<m<<":"<<s<<" "<< mer;

}

};

class Time24

{

protected:

int h,m,s;

public:

void gettime()

{

cout<<"Enter Hour :- ";

cin>>h;

cout<<"Enter Minute :- ";

cin>>m;

cout<<"Enter Second :- ";

cin>>s;

}

void display()

{

gettime();

cout<<"Time :-\n";

cout<<h<<":"<<m<<":"<<s;

}

};

void main()

{

Time12 t1;

Time24 t2;

clrscr();

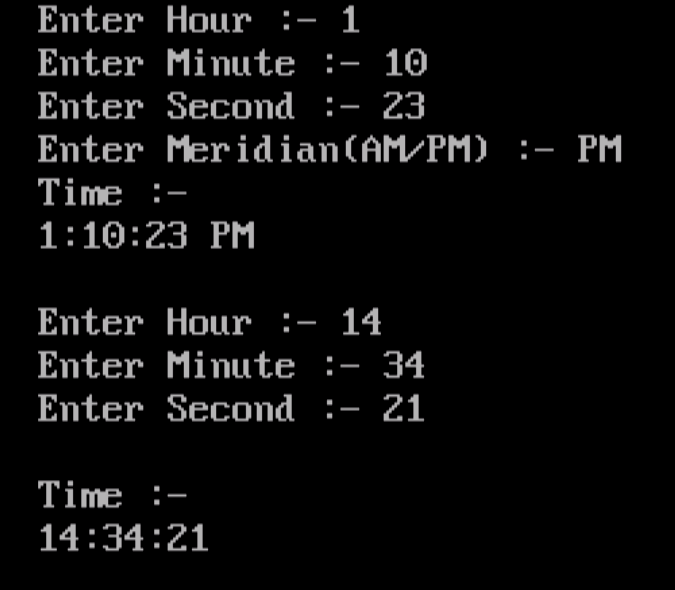
t1.display();

t2.display();

getch();

}

**Output**

****

**Question 7**

**Create two classes DM and DB which store the values of distance. DM stores distance in meters and centimeters. DB stores distances in feet and inches. Write a program that can read values for the class object and add one object of DM with another object of DB. Use a friend function to carry out the addition operation and this function will display answer in meter and centimeters.**

**Solution**

#include<iostream.h>

#include<conio.h>

class DB;

class DM

{

protected:

int m,cm;

public:

DM()

{

cout<<"Enter Meter :- ";

cin>>m;

cout<<"Enter Centimeter :- ";

cin>>cm;

}

friend void add(DM m,DB b);

};

class DB

{

protected:

int f,i;

public:

DB()

{

cout<<"Enter Feet :- ";

cin>>f;

cout<<"Enter Inches :- ";

cin>>i;

}

friend void add(DM m,DB b);

};

void add(DM m,DB b)

{

float m1,cm;

cm=m.cm+(b.i\*2.54)+(b.f\*30.48);

m1=m.m+(int)(cm/100);

cm=(int)cm%100;

cout<<"Meter :- "<<m1<<endl;

cout<<"Centimeter :- "<<cm<<endl;

}

void main()

{

clrscr();

DM m;

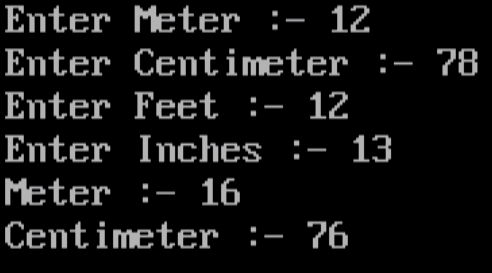
DB b;

add(m,b);

getch();

}

**Output**



**Question 8**

**Write a program to maintain a telephone directory use add() and Show() methods to add new entries and display the telephone numbers of a person when the name of the person is given.**

**Solution**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class Telephone

{

protected:

char name[20][20];

long int tno[20];

int i,j;

public:

Telephonr()

{

i=0;

}

void add()

{

cout<<"Enter Name :- ";

cin>>name[i];

cout<<"Enter Telephone Number :- ";

cin>>tno[i];

i++;

}

void show(char nm[20])

{

for(j=0;j<i;j++)

{

if(strcmp(nm,name[j])==0)

{

cout<<"Telephone Number :- "<<tno[j]<<endl;

break;

}

}

if(j==i)

cout<<"Name Not Found..."<<endl;

}

};

void main()

{

int ch;

char nm[20];

Telephone t;

clrscr();

do

{

cout<<"\n1.Add\n2.Show\nEnter Choice :- ";

cin>>ch;

switch(ch)

{

case 1:

t.add();

break;

case 2:

cout<<"Enter Name To Search Number :- ";

cin>>nm;

t.show(nm);

break;

case 3:

break;

default:

cout<<"Enter Correct Choice...";

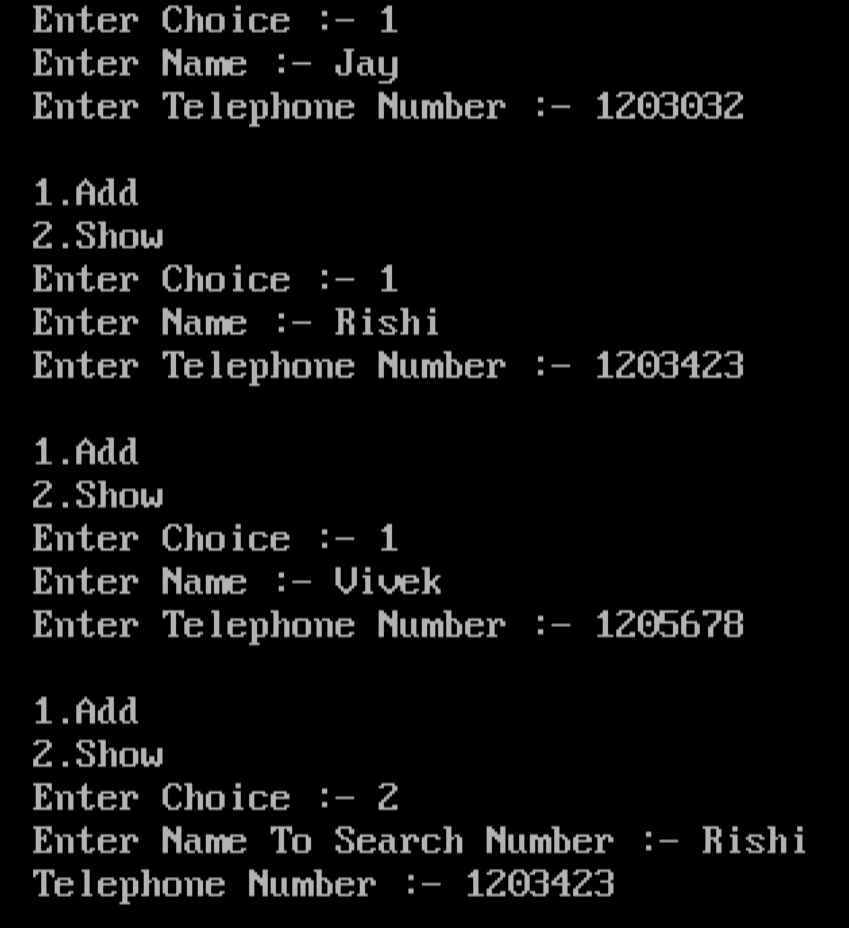
}

}while(ch!=3);

getch();

}

**Output**



**Question 9**

**Create a base class shape use the class two store double type value that could be used to compare the area. A drive to specific classes called triangle and rectangle. From the base shape and a member in get data to the base class to initialize base data member and another function display area**

**Solution**

#include<iostream.h>

#include<conio.h>

class Shape

{

protected:

double w,h;

public:

virtual void getdata()=0;

virtual void area()=0;

};

class Triangle:public Shape

{

public:

void getdata()

{

cout<<"Enter Values For Triangle :-\n";

cout<<"Enter Height :- ";

cin>>h;

cout<<"Enter Width :- ";

cin>>w;

}

void area()

{

cout<<"Area Of Triangle :- "<<(h\*w)/2;

}

};

class Rectangle:public Shape

{

public:

void getdata()

{

cout<<"\nEnter Values For Rectangle :-\n";

cout<<"Enter Length :- ";

cin>>h;

cout<<"Enter Width :- ";

cin>>w;

}

void area()

{

cout<<"Area Of Rectangle :- "<<h\*w;

}

};

void main()

{

clrscr();

Shape \*t,\*r;

t=new Triangle();

r=new Rectangle();

t->getdata();

t->area();

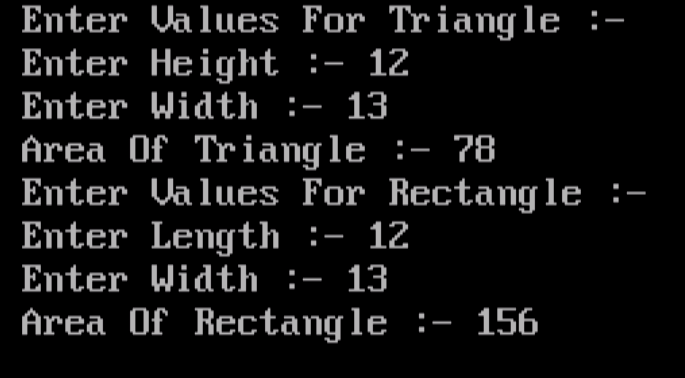
r->getdata();

r->area();

getch();

}

**Output**



**Question 10**

**Write Program to implement Stack Operations like PUSH, POP, PEEP, UPDATE and DISPLAY using class and object.**

**Solution**

#include<iostream.h>

#include<conio.h>

const int n=5;

int s[n],top=-1;

class Stack

{

public:

void push(int data)

{

if(top==n-1)

{

cout<<"Stack is Overflow..."<<endl;

return;

}

top++;

s[top]=data;

}

int pop()

{

if(top==-1)

{

cout<<"Stack is Underflow..."<<endl;

return 0;

}

int data=s[top];

top--;

return data;

}

void peep()

{

int loc;

cout<<"Enter Location From Top :- ";

cin>>loc;

if(top-loc+1<0)

{

cout<<"There is no value..."<<endl;

return;

}

cout<<"Value at "<<loc<<" from Top :- "<<s[top-loc+1]<<endl;

}

void update()

{

int loc,data;

cout<<"Enter Location From Top :- ";

cin>>loc;

if(top-loc+1<0)

{

cout<<"There is no value..."<<endl;

return;

}

cout<<"Value at "<<loc<<" from Top :- "<<s[top-loc+1]<<endl;

cout<<"Enter New Value :- ";

cin>>data;

s[top-loc+1]=data;

cout<<"Value Updated..."<<endl;

}

void display()

{

int i;

if(top==-1)

{

cout<<"Stack is Underflow..."<<endl;

return;

}

for(i=top;i>=0;i--)

cout<<s[i]<<endl;

}

};

void main()

{

int ch,data;

Stack s1;

clrscr();

do

{

cout<<"1.Push\n2.Pop\n3.Peep\n4.Update\n5.Display\n6.Exit\n";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Value :- ";

cin>>data;

s1.push(data);

break;

case 2:

cout<<"Removed Value :- "<<s1.pop()<<endl;

break;

case 3:

s1.peep();

break;

case 4:

s1.update();

break;

case 5:

s1.display();

break;

case 6:

break;

default:

cout<<"Enter Correct Choice..."<<endl;

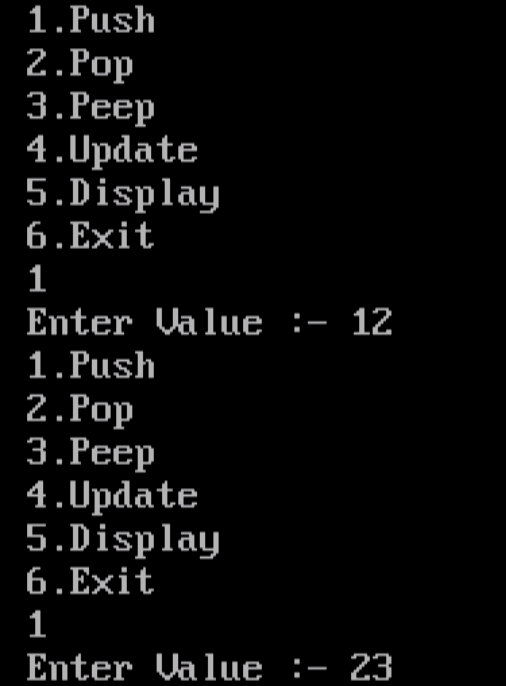
}

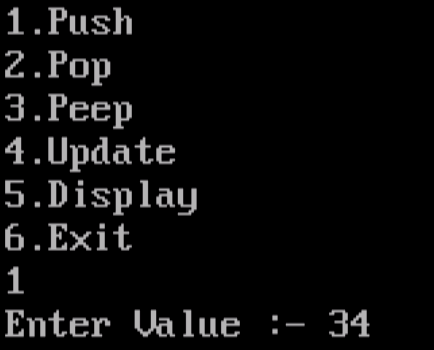
}while(ch!=6);

getch();

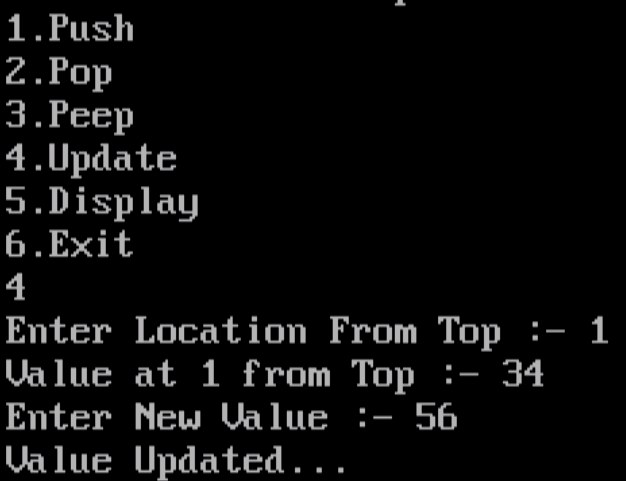
}

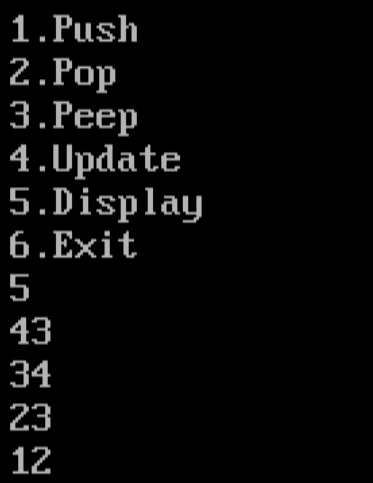
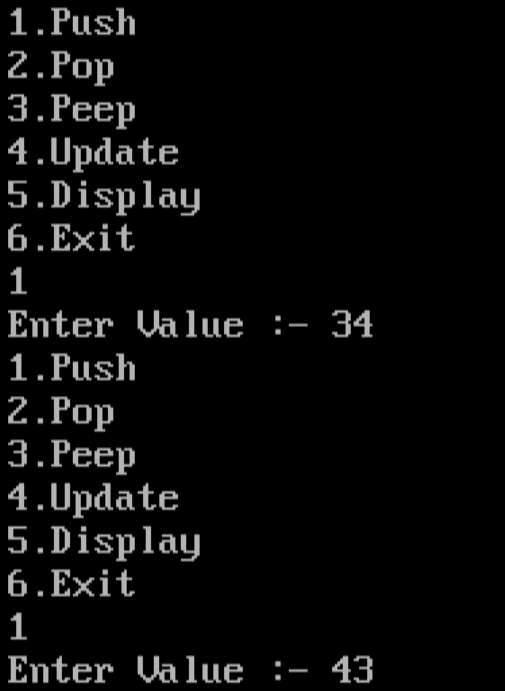
**Output**











**Question 11**

**Write Program to convert Infix to Postfix Expression using class and object.**

**Solution**

#include<iostream.h>

#include<conio.h>

const int n=20;

char s[n];

int top=-1;

class Stack

{

public:

void push(char data)

{

top++;

s[top]=data;

}

char pop()

{

char data=s[top];

top--;

return data;

}

int priority(char op)

{

switch(op)

{

case '^':

return 3;

case '\*':case '/':

return 2;

case '+':case '-':

return 1;

default:

return 0;

}

}

};

void main()

{

char in[n],post[n];

int i,j=0;

Stack st;

clrscr();

cout<<"Enter Infix Expression :- ";

cin>>in;

for(i=0;in[i]!='\0';i++)

{

switch(in[i])

{

case '(':

st.push(in[i]);

break;

case '^':case '\*':case '/':case '+':case '-':

while(st.priority(s[top])>=st.priority(in[i]))

post[j++]=st.pop();

st.push(in[i]);

break;

case ')':

while(s[top]!='(')

post[j++]=st.pop();

st.pop();

break;

default:

post[j++]=in[i];

}

}

while(top>-1)

post[j++]=st.pop();

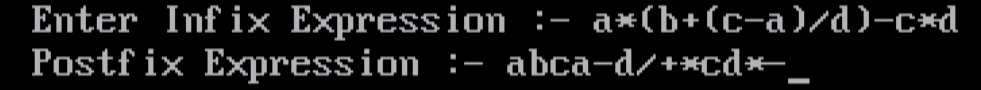
post[j]='\0';

cout<<"Postfix Expression :- "<<post;

getch();

}

**Output**



**Question 12**

**Write Program to convert Infix to Prefix Expression using class and object**

**Solution**

#include<iostream.h>

#include<conio.h>

#include<string.h>

const int n=20;

char s[n];

int top=-1;

class Stack

{

public:

void push(char data)

{

top++;

s[top]=data;

}

char pop()

{

char data=s[top];

top--;

return data;

}

int priority(char op)

{

switch(op)

{

case '^':

return 3;

case '\*':case '/':

return 2;

case '+':case '-':

return 1;

default:

return 0;

}

}

};

void main()

{

char in[n],pre[n];

int i,j=0;

Stack st;

clrscr();

cout<<"Enter Infix Expression :- ";

cin>>in;

strrev(in);

for(i=0;in[i]!='\0';i++)

{

switch(in[i])

{

case ')':

st.push(in[i]);

break;

case '^':case '\*':case '/':case '+':case '-':

while(st.priority(s[top])>=st.priority(in[i]))

pre[j++]=st.pop();

st.push(in[i]);

break;

case '(':

while(s[top]!=')')

pre[j++]=st.pop();

st.pop();

break;

default:

pre[j++]=in[i];

}

}

while(top>-1)

pre[j++]=st.pop();

pre[j]='\0';

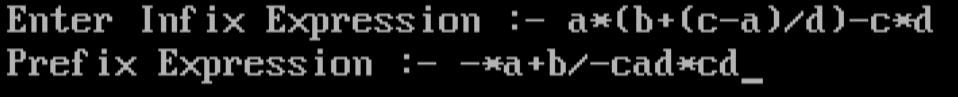
strrev(pre);

cout<<"Prefix Expression :- "<<pre;

getch();

}

**Output**



**Question 13**

**Write Program to implement Simple Queue Operations like Insert, Delete and Display.**

**Solution**

#include<iostream.h>

#include<conio.h>

const n=5;

int f=-1,r=-1,q[n];

class Queue

{

public:

void enqueue(int data)

{

if(r==n-1)

{

cout<<"Queue is Overflow..."<<endl;

return;

}

r++;

q[r]=data;

if(f==-1)

f++;

}

int dequeue()

{

int data;

if(f==-1)

{

cout<<"Queue is Underflow..."<<endl;

return 0;

}

data=q[f];

if(f==r)

f=r=-1;

else

f++;

return data;

}

void display()

{

int i;

if(f==-1)

{

cout<<"Queue is Underflow..."<<endl;

return;

}

for(i=f;i<=r;i++)

cout<<q[i]<<"\t";

}

};

void main()

{

int ch,data;

Queue q1;

clrscr();

do

{

cout<<"\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Value :- ";

cin>>data;

q1.enqueue(data);

break;

case 2:

cout<<"Removed Data :- "<<q1.dequeue()<<endl;

break;

case 3:

q1.display();

break;

case 4:

break;

default:

cout<<"Enter Correct Choice..."<<endl;

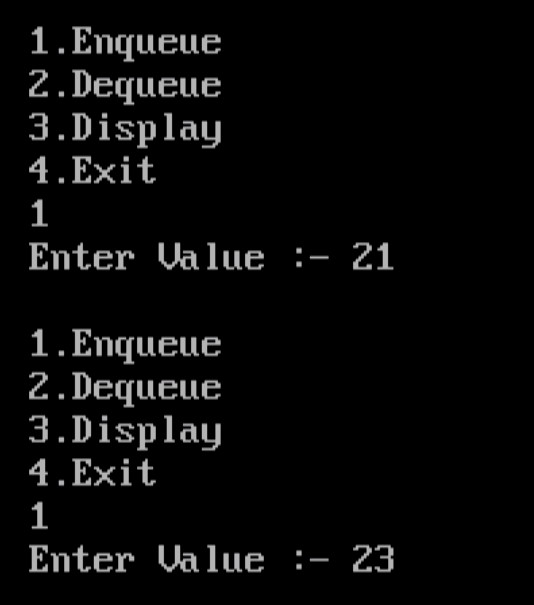
}

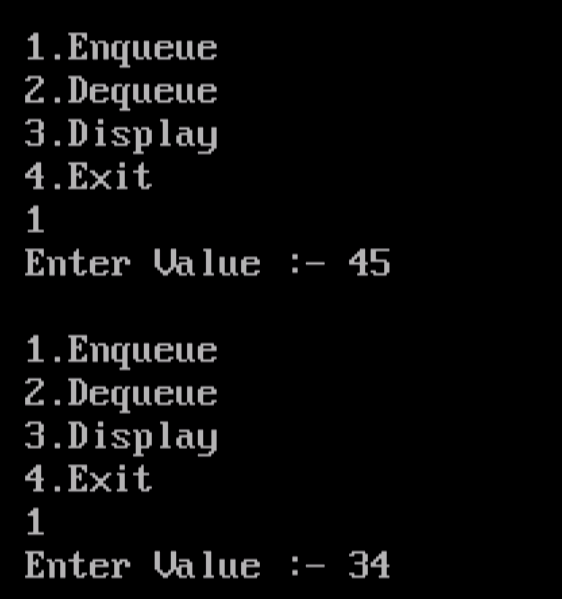
}while(ch!=4);

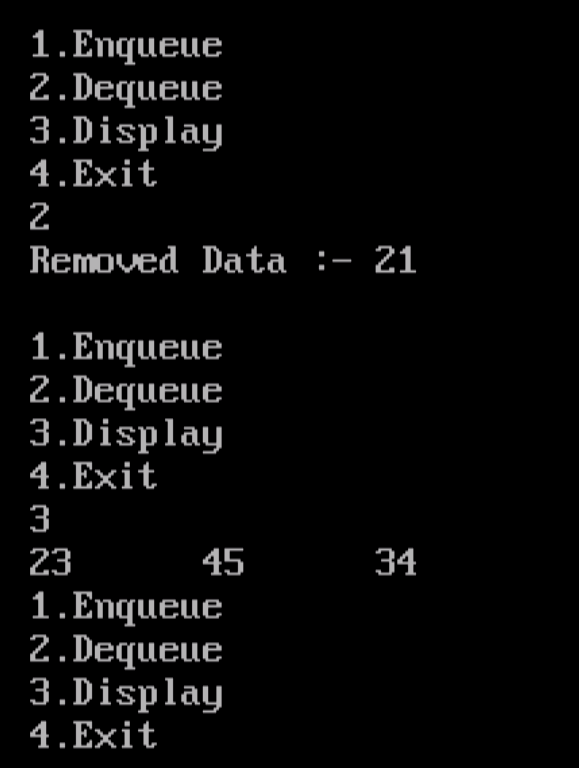
getch();

}

**Output**







**Question 14**

**Write Program to implement Circular Queue Operations like Insert, Delete and Display using class and object.**

**Solution**

#include<iostream.h>

#include<conio.h>

const int n=5;

int f=-1,r=-1,q[n];

class Queue

{

public:

void enqueue(int data)

{

if((r==n-1 && f==0)||r+1==f)

{

cout<<"Queue is Overflow..."<<endl;

return;

}

if(r==n-1)

r=0;

else

r++;

q[r]=data;

if(f==-1)

f++;

}

int dequeue()

{

int data;

if(f==-1)

{

cout<<"Queue is Underflow..."<<endl;

return 0;

}

data=q[f];

if(f==r)

f=r=-1;

else if(f==n-1)

f=0;

else

f++;

return data;

}

void display()

{

int i;

if(f==-1)

{

cout<<"Queue is Underflow..."<<endl;

return;

}

if(f<=r)

{

for(i=f;i<=r;i++)

{

cout<<q[i]<<"\t";

}

}

else

{

for(i=f;i<n;i++)

cout<<q[i]<<"\t";

for(i=0;i<=r;i++)

cout<<q[i]<<"\t";

}

}

};

void main()

{

int ch,data;

Queue q1;

clrscr();

do

{

cout<<"\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit\n";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Value :- ";

cin>>data;

q1.enqueue(data);

break;

case 2:

cout<<"Removed Data :- "<<q1.dequeue()<<endl;

break;

case 3:

q1.display();

break;

case 4:

break;

default:

cout<<"Enter Correct Choice..."<<endl;

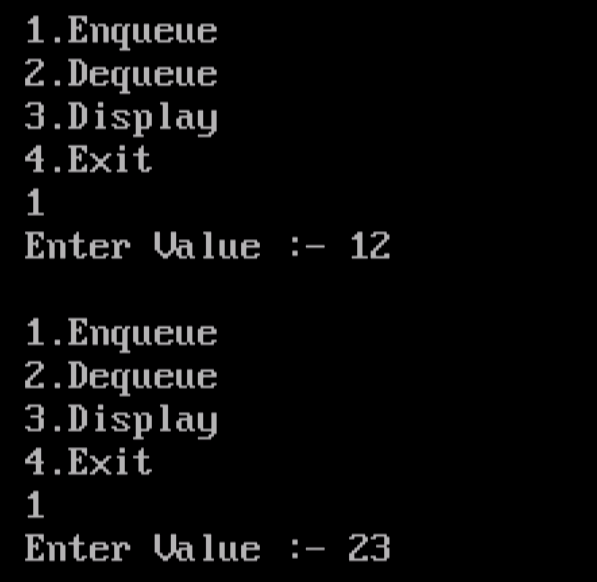
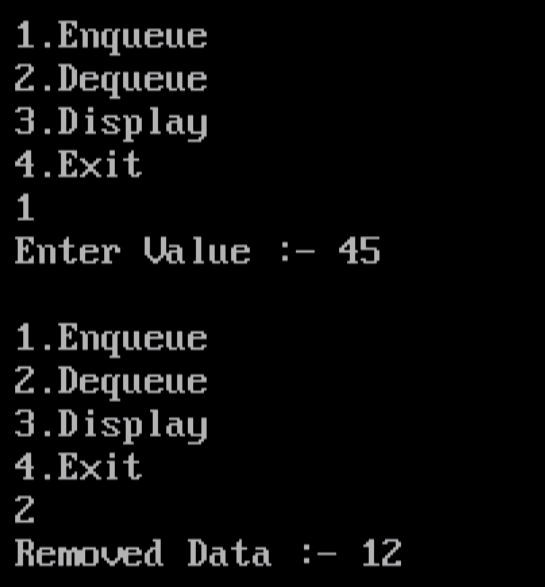
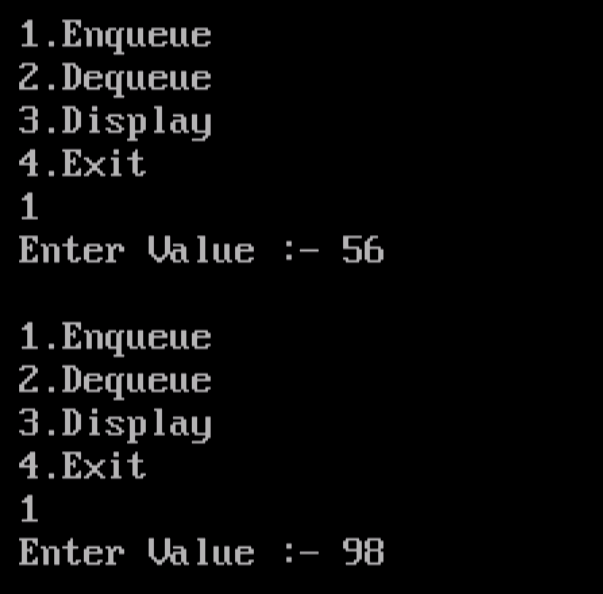
}

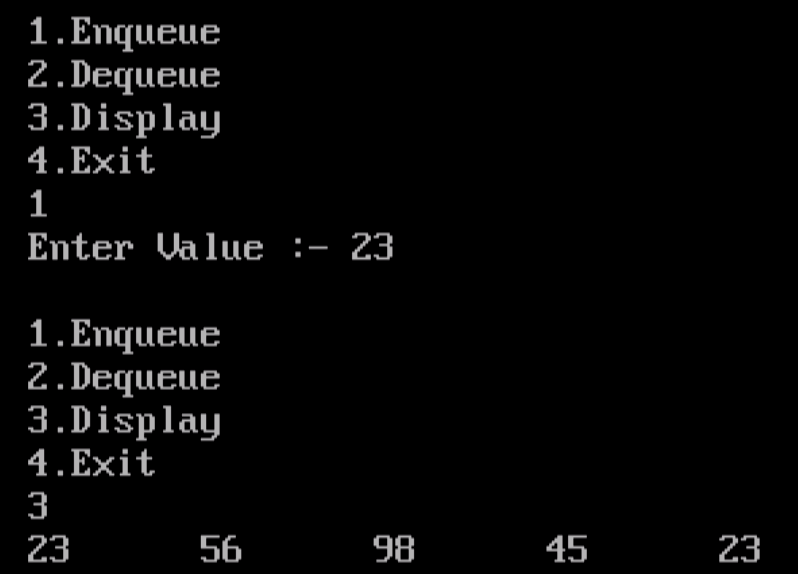
}while(ch!=4);

getch();

}

**Output**





**Question 15**

**Write Program to implement Double Ended Queue Operations like Insert, Delete and Display using class and object(To Perform Input Restricted).**

**Solution**

#include<iostream.h>

#include<conio.h>

const int n=5;

int dq[n],f=-1,r=-1;

class Queue

{

public:

void insert\_front(int data)

{

if(f==0)

{

cout<<"Dequeue is Overflow"<<endl;

return;

}

if(f==-1)

f=r=n-1;

else

f--;

dq[f]=data;

}

void insert\_rear(int data)

{

if(r==n-1)

{

cout<<"Dequeue is Overflow"<<endl;

return;

}

r++;

dq[r]=data;

if(f==-1)

f++;

}

int delete\_front()

{

if(f==-1)

{

cout<<"Dequeue is Underflow"<<endl;

return 0;

}

int data=dq[f];

dq[f]=0;

if(f==r)

f=r=-1;

else

f++;

return data;

}

int delete\_rear()

{

if(r==-1)

{

cout<<"Dequeue is Underflow"<<endl;

return 0;

}

int data=dq[r];

dq[r]=0;

if(f==r)

f=r=-1;

else

r--;

return data;

}

void display()

{

int i;

if(f==-1)

{

cout<<"Dequeue is Underflow..."<<endl;

return;

}

for(i=0;i<n;i++)

cout<<dq[i]<<"\t";

}

};

void main()

{

int ch,data;

char rside,side;

Queue q1;

clrscr();

cout<<"\nEnter Restricted Side(L/R) :- ";

cin>>rside;

do

{

cout<<"\n\n1.Insert\n2.Delete\n3.Display\n4.Exit\nEnter Choice :- ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Value :- ";

cin>>data;

if(rside=='L')

q1.insert\_rear(data);

else

q1.insert\_front(data);

break;

case 2:

int del;

cout<<"Enter Side To Delete Data(L/R) :- ";

cin>>side;

if(side=='L')

del=q1.delete\_front();

else

del=q1.delete\_rear();

if(del!=0)

cout<<"Deleted Value :- "<<del<<endl;

break;

case 3:

q1.display();

break;

case 4:

cout<<"Exitting...";

break;

default:

cout<<"Enter Correct Choice...";

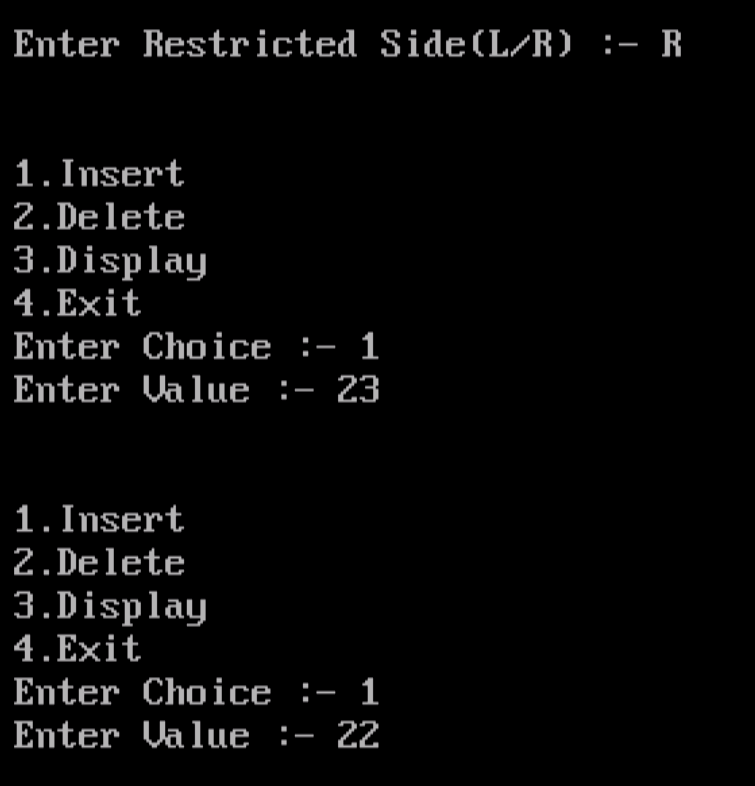
}

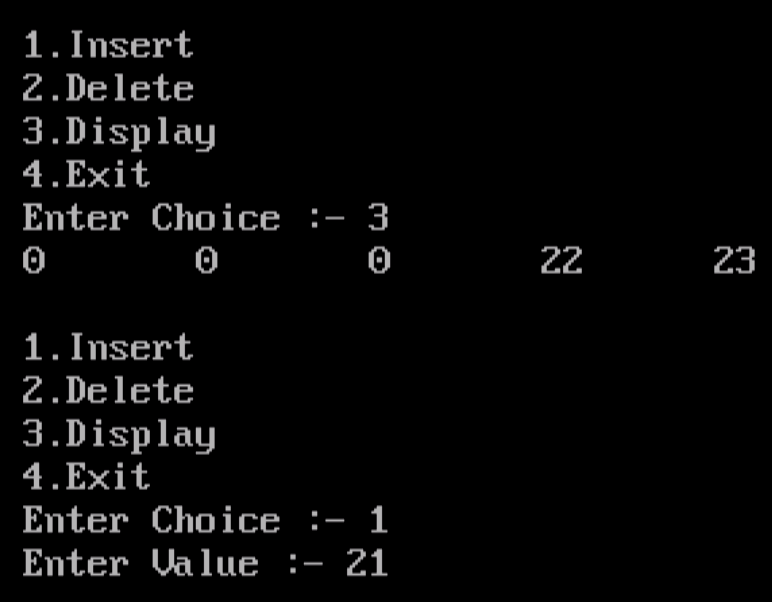
}while(ch!=4);

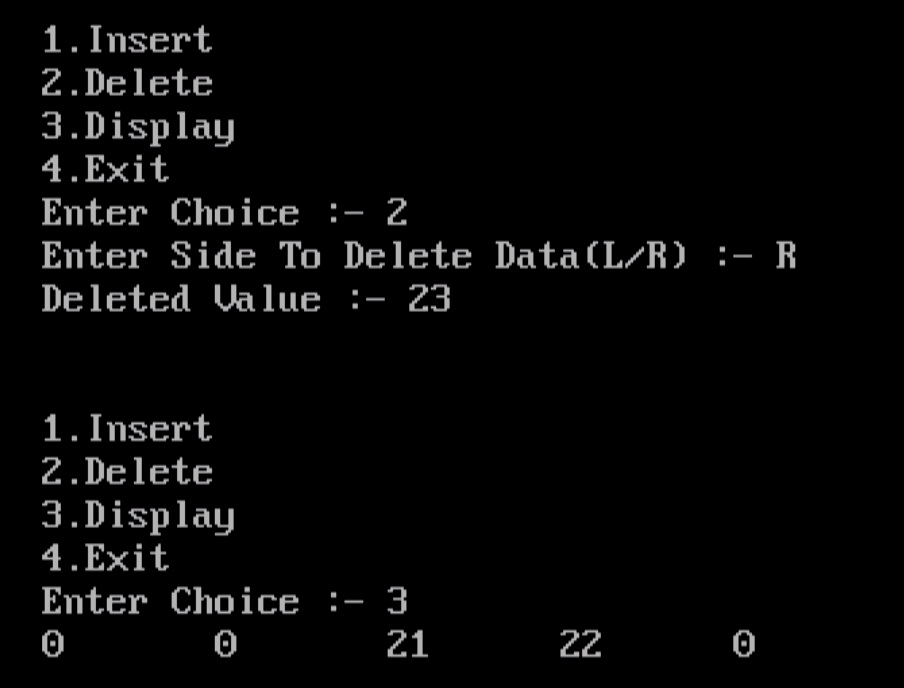
getch();

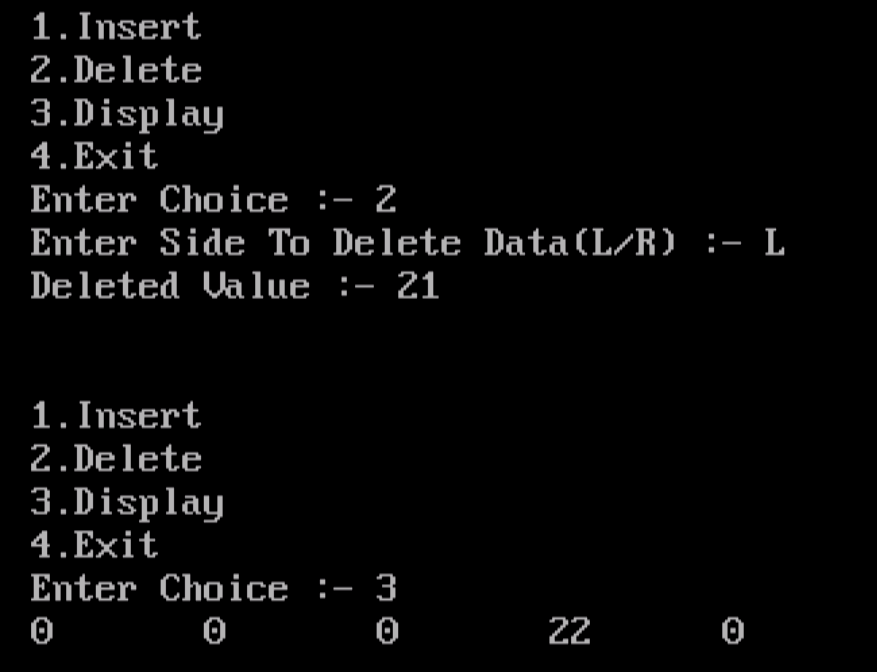
}

**Output**









**Question 16**

**Write Program to implement Double Ended Queue Operations like Insert, Delete and Display using class and object(To Perform Output Restricted).**

**Solution**

#include<iostream.h>

#include<conio.h>

const int n=5;

int dq[n],f=-1,r=-1;

class Queue

{

public:

void insert\_front(int data)

{

if(f==0)

{

cout<<"Dequeue is Overflow"<<endl;

return;

}

if(f==-1)

f=n-1;

else

f--;

dq[f]=data;

}

void insert\_rear(int data)

{

if(r==n-1)

{

cout<<"Dequeue is Overflow"<<endl;

return;

}

r++;

dq[r]=data;

}

int delete\_front()

{

if(f==-1)

{

cout<<"Dequeue is Underflow"<<endl;

return 0;

}

int data=dq[f];

dq[f]=0;

if(f==r)

f=r=-1;

else

f++;

return data;

}

int delete\_rear()

{

if(r==-1)

{

cout<<"Dequeue is Underflow"<<endl;

return 0;

}

int data=dq[r];

dq[r]=0;

if(f==r)

f=r=-1;

else

r--;

return data;

}

void display()

{

int i;

if(f==-1)

{

cout<<"Dequeue is Underflow..."<<endl;

return;

}

for(i=0;i<n;i++)

cout<<dq[i]<<"\t";

}

};

void main()

{

int ch,data;

char rside,side;

Queue q1;

clrscr();

cout<<"\nEnter Restricted Side(L/R) :- ";

cin>>rside;

do

{

cout<<"\n\n1.Insert\n2.Delete\n3.Display\n4.Exit\nEnter Choice :- ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter Value :- ";

cin>>data;

cout<<"Enter Side To Delete Data(L/R) :- ";

cin>>side;

if(side=='R')

q1.insert\_rear(data);

else

q1.insert\_front(data);

break;

case 2:

int del;

if(rside=='R')

del=q1.delete\_front();

else

del=q1.delete\_rear();

if(del!=0)

cout<<"Deleted Value :- "<<del<<endl;

break;

case 3:

q1.display();

break;

case 4:

cout<<"Exitting...";

break;

default:

cout<<"Enter Correct Choice...";

}

}while(ch!=4);

getch();

}

**Output**

