Q1. Write a C++ program to calculate the area of triangle, rectangle and circle using constructor overloading. The program should be menu driven.

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

class Area {
public:
    Area(double base, double height) {
        double area = (base * height) / 2;
        cout << "Area of Triangle: " << area << endl;
    }

    Area(double length, double width) {
        double area = length * width;
        cout << "Area of Rectangle: " << area << endl;
    }

    Area(double radius) {
        double area = 3.14 * radius * radius;
        cout << "Area of Circle: " << area << endl;
    }
};

void main() {
    int choice;
    double a, b;

    clrscr();
    cout << "\nEnter your choice:\n";
    cout << "Choice 1: Area of Triangle\n";
    cout << "Choice 2: Area of Rectangle\n";
    cout << "Choice 3: Area of Circle\n";
    cout << "Choice 4: Exit\n";

    while (1) {
        cout << "\nChoice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter base and height of Triangle: ";
```

```
                cin >> a >> b;
                Area triangle(a, b);
                break;

            case 2:
                cout << "Enter length and width of Rectangle: ";
                cin >> a >> b;
                Area rectangle(a, b);
                break;

            case 3:
                cout << "Enter radius of Circle: ";
                cin >> a;
                Area circle(a);
                break;

            case 4:
                cout << "Exit the program\n";
                getch();
                return;

            default:
                cout << "Invalid choice.\n";
                break;
        }
    }
}
```

Q2. Create a C++ class for player object with the following attributes player no., name, number of matches and number of goals done in each match. The number of matches varies for each player. Write parameterized constructor which initializes player no., name, number of subjects and creates array for number of goals and number of matches dynamically.

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

class Player {
```

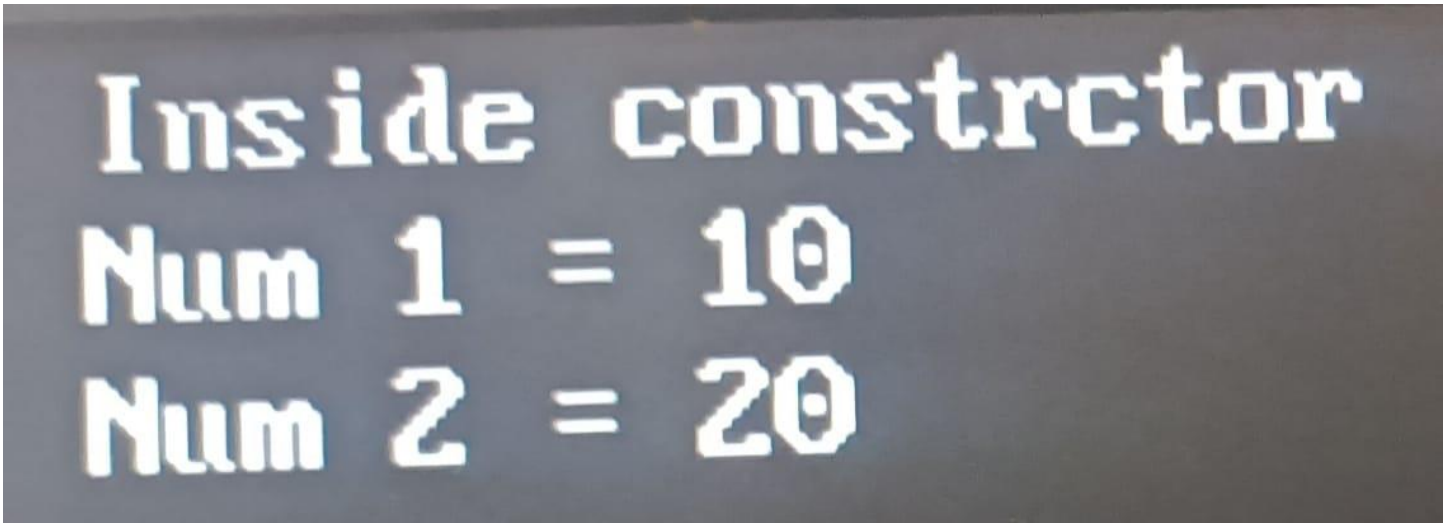
```
private:
    int playerNo;
    char name[50];
    int numMatches;
    int * goals;
public:

    Player(int pNo, const char * pName, int matches) {
    playerNo = pNo;
    strcpy(name, pName);
    numMatches = matches;
    goals = new int[numMatches];
    }

    ~Player() {
    delete[] goals;
    }

    void setGoals() {
    cout << "Enter goals for each match:\n";
    for (int i = 0; i < numMatches; i++) {
        cout << "Match " << (i + 1) << ": ";
        cin >> goals[i];
    }
    }

    void display() {
    cout << "\nPlayer No: " << playerNo;
    cout << "\nName: " << name;
    cout << "\nNumber of Matches: " << numMatches;
    cout << "\nGoals in each match:\n";
    for (int i = 0; i < numMatches; i++) {
        cout << "Match " << (i + 1) << ": " << goals[i] << "\n";
    }
    }
};

void main() {
    clrscr();
    int playerNo, matches;
    char name[50];
```

```
    cout << "Enter player number: ";
    cin >> playerNo;
    cout << "Enter player name: ";
    cin >> name;
    cout << "Enter number of matches: ";
    cin >> matches;

    Player player(playerNo, name, matches);
    player.setGoals();
    player.display();

    getch();
}
```

```
Enter player number: 1
Enter player name: abc
Enter number of matches: 5
Enter goals for each match:
Match 1: 24
Match 2: 45
Match 3: 43
Match 4: 64
Match 5: 24

Player No: 1
Name: abc
Number of Matches: 5
Goals in each match:
Match 1: 24
Match 2: 45
Match 3: 43
Match 4: 64
Match 5: 24
```

Q3. Write a C++ program to demonstrate the destructor.

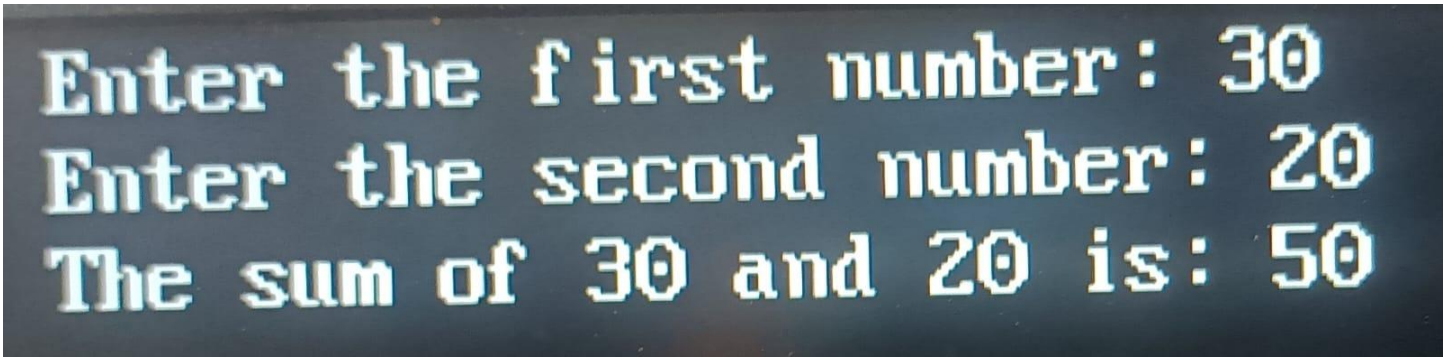
```
#include<iostream.h>
#include<conio.h>

class demo
{
	int num1,num2;
	public:
	demo(int n1, int n2)
	{
	cout<<"Inside constrctor"<<endl;
	num1 = n1;
	num2 = n2;
	}

	void display()
	{
	cout<<"Num 1 = "<<num1<<endl;
	cout<<"Num 2 = "<<num2<<endl;
	}
	~demo()
	{
	cout<<"Inside destrctor";
	}
};

int main() {

	clrscr();
	demo obj1(10,20);
	obj1.display();
	getch();
}
```



Q5. Write a C++ program to add two numbers using single inheritance. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.

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

class Base
{
    protected:
        int num1, num2;

    public:

        void getNumbers() {
            cout << "Enter the first number: ";
            cin >> num1;
            cout << "Enter the second number: ";
            cin >> num2;
        }
};

class Derived : public Base
{
    public:

        void displaySum() {
            int sum = num1 + num2;
            cout << "The sum of " << num1 << " and " << num2 << " is: " << sum << endl;
```

```
        }
};

void main() {
    clrscr();

    Derived obj;

    obj.getNumbers();
    obj.displaySum();

    getch();
}
```



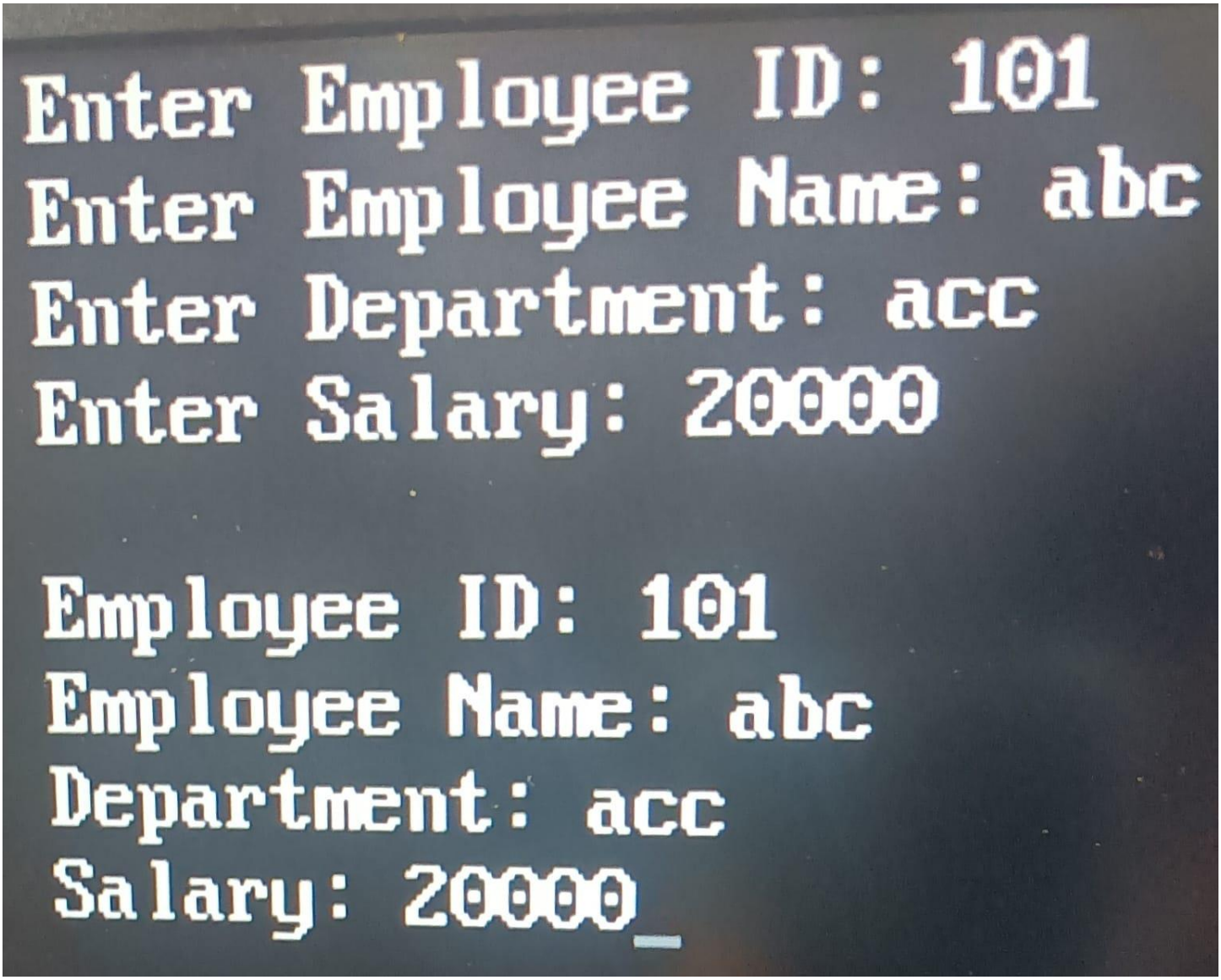
Q11. . Write a C++ Program to Maintain Employee Database using Virtual class.

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

class Employee
{
    protected:
        int emp_id;
        char emp_name[50];
    public:
        Employee()
        {
            emp_id = 0;
            strcpy(emp_name, " ");
        }
```

```
        void getEmployeeData()
        {
            cout << "Enter Employee ID: ";
            cin >> emp_id;
            cout << "Enter Employee Name: ";
            cin >> emp_name;
        }

        void displayEmployeeData()
        {
            cout << "\nEmployee ID: " << emp_id;
            cout << "\nEmployee Name: " << emp_name;
        }
};

class Department : virtual public Employee
{
    protected:
        char department[50];
    public:
        Department()
        {
            strcpy(department, " ");
        }

        void getDepartmentData()
        {
            cout << "Enter Department: ";
            cin >> department;
        }

        void displayDepartmentData()
        {
            cout << "\nDepartment: " << department;
        }
};

class Salary : virtual public Employee
{
    protected:
        double salary;
    public:
```

```
        Salary()
        {
            salary = 0.0;
        }
        void getSalaryData()
        {
            cout << "Enter Salary: ";
            cin >> salary;
        }
        void displaySalaryData()
        {
            cout << "\nSalary: " << salary;
        }
};

class EmployeeDatabase : public Department, public Salary
{
    public:
        EmployeeDatabase()
        : Employee(), Department(), Salary() {}
        void getAllData()
        {
            getEmployeeData();
            getDepartmentData();
            getSalaryData();
        }

        void displayAllData()
        {
            displayEmployeeData();
            displayDepartmentData();
            displaySalaryData();
        }
};

void main()
{
    clrscr();

    EmployeeDatabase emp;

    emp.getAllData();
```

```
    emp.displayAllData();

    getch();
}
```



q12. With the help of c++ class and object, stack will be implemented with following operations 1. Push Operation 2. Pop Operations 3. Peep Operation 4. Update Operation 5. Display Operation

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

```
#include<conio.h>
#define n 5

int stack[n], top = -1;
void push();
void pop();
void peep();
void update();
void display();

int main() {
    int ch;
    clrscr();
    while(1) {
        cout<<"\n Enter your choice \n";
        cout<<"\n Enter 1 : for push ";
        cout<<"\n Enter 2 : for pop ";
        cout<<"\n Enter 3 : for peep ";
        cout<<"\n Enter 4 : for update ";
        cout<<"\n Enter 5 : for display ";
        cout<<"\n Enter 6 : for exit ";
        cin>>ch;

     switch(ch) {
        case 1:
        push();
        break;
        case 2:
        pop();
        break;
        case 3:
        peep();
        break;
        case 4:
        update();
        break;
        case 5:
        display();
        break;
        case 6:
        cout<<"Exit";
        return 0;
```

```
        default:
        cout<<"Wrong Choice";
    }
    }
}

void push() {
    int data;

    cout<<"\n Enter the data ";
    cin>>data;
    if(top==n-1)
    {
    cout<<("Overflow");
    }
    else
    {
    top++;
    stack[top] = data;
    cout<<"\n Data inserted";
    }
}

void pop() {
    int item;
    if(top==-1)
    {
    cout<<"Underflow";
    }
    else
    {item = stack[top];
    top--;
    cout<<item;
    }
}

void peep() {
    if(top == -1) {
        cout<<"stack is empty\n";
    }
    else
    {
```

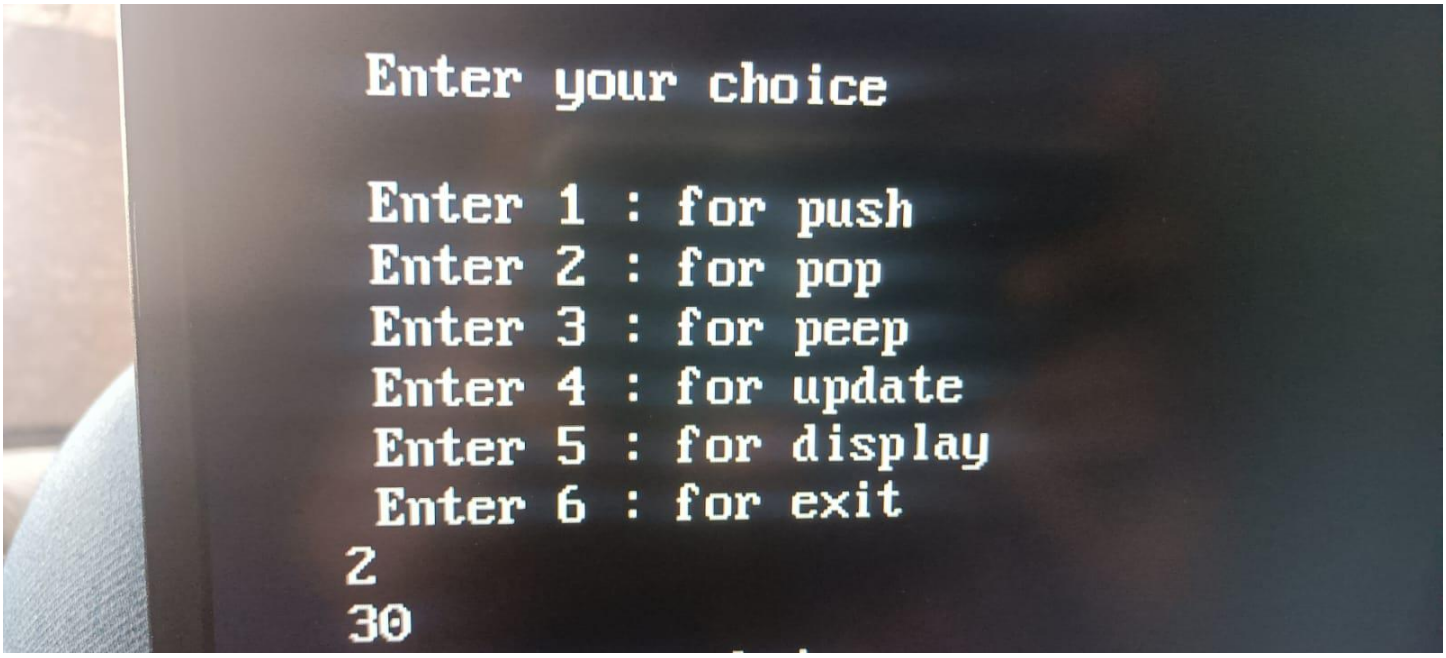
```
        cout<<"Top element : "<<stack[top]<<endl;
    }
}

void update() {
    if(top == -1) {
        cout<<"Stack is empty\n";
    }
    else
    {
    int index, newdata;
    cout<<"Enter index to update (0 to " <<top << "): " ;
    cin>>index;
    if(index<0 || index > top) {
        cout<<"Invalid index\n";
    }
    else
    {
        cout<<"Enter new data : ";
        cin>>newdata;
        stack[index] = newdata;
        cout<<"Data update to index "<< index <<" to " << newdata <<endl;
    }
    }
}

void display() {
    int i;
    for(i=0;i<=top;i++)
    cout<<stack[i]<<endl;
}
```

```
Enter your choice

Enter 1 : for push
Enter 2 : for pop
Enter 3 : for peep
Enter 4 : for update
Enter 5 : for display
Enter 6 : for exit
1

Enter the data 10
```

```
Enter your choice

Enter 1 : for push
Enter 2 : for pop
Enter 3 : for peep
Enter 4 : for update
Enter 5 : for display
Enter 6 : for exit
4
Enter index to update (0 to 2): 1
Enter new data : 50
Data update to index 1 to 50
```

```
Enter your choice

Enter 1 : for push
Enter 2 : for pop
Enter 3 : for peep
Enter 4 : for update
Enter 5 : for display
Enter 6 : for exit
3
Top element : 30
```

```
Enter 1 : for push
Enter 2 : for pop
Enter 3 : for peep
Enter 4 : for update
Enter 5 : for display
Enter 6 : for exit
5
10
20
30
```

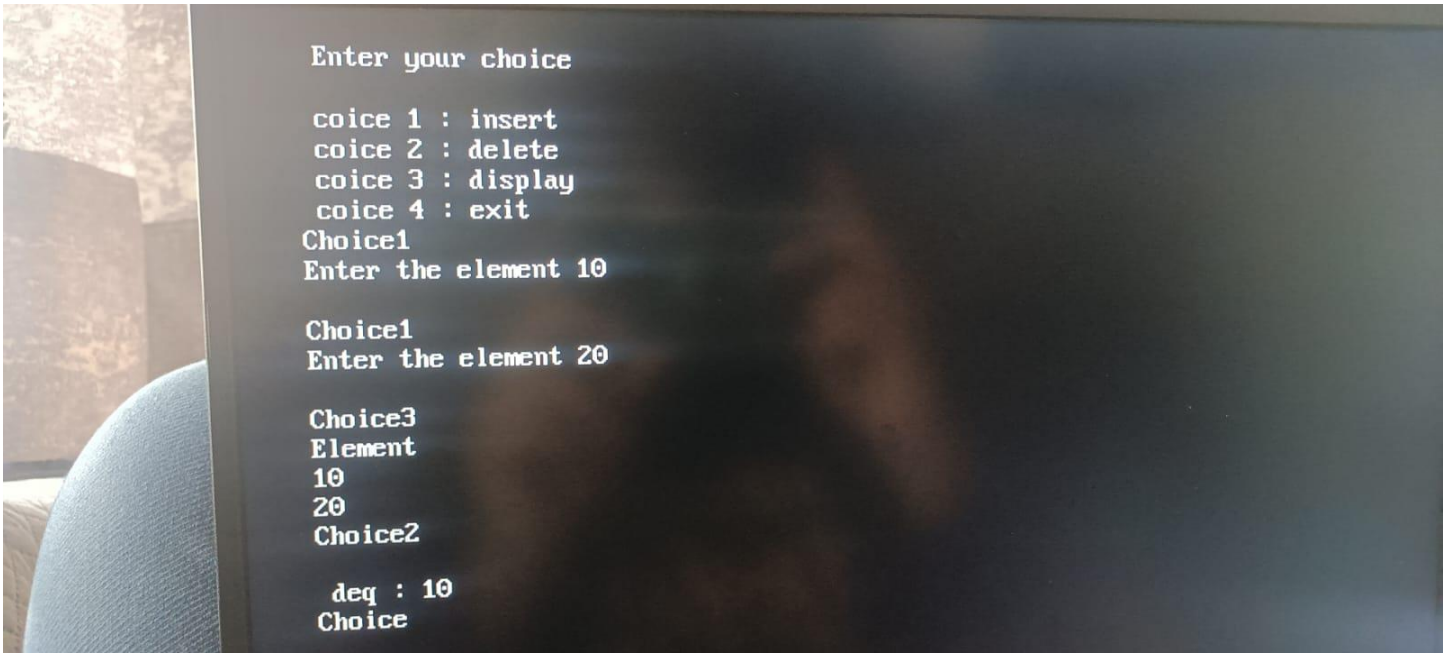


Q13. Write a C++ Program to Implement Simple Queue.(using class and object)

```
#include<iostream.h>
#include<conio.h>
#define max 5

int que[max],front=-1,rear=-1;
void enq();
void deq();
void dis();

void main() {
    int ch;
    clrscr();
    cout<<"\n Enter your choice \n";
    cout<<"\n coice 1 : insert ";
    cout<<"\n coice 2 : delete ";
    cout<<"\n coice 3 : display ";
    cout<<"\n coice 4 : exit ";

    while(1) {
    cout<<"\nChoice";
    cin>>ch;
    switch(ch) {
```

```
        case 1:
        enq();
        break;
        case 2:
        deq();
        break;
        case 3:
        dis();
        break;
        case 4:
        cout<<"Exit";
        getch();
        default:
        cout<<"\nWrong choice\n";
    }
    }
}

void enq() {
    int data;
    if(rear==max-1) {
    cout<<"Overflow";
    }
    else
    {
    if(front==-1);
    }
    front=0;
    {
    cout<<"Enter the element ";
    cin>>data;
    rear++;
    que[rear]=data;
    }
}

void deq() {
    if(front==-1 || front>rear)
    {
    cout<<"\nUnderflow\n";
    }
    else
```

```
    {
    cout<<"\n deq : "<<que[front];
    front++;
    }
}

void dis() {
    int i;
    if(front==-1)
    {
    cout<<"\n queue is empty";
    }
    else
    {
    cout<<"Element";
    }
    for(i=front;i<=rear;i++)
    {
    cout<<"\n"<<que[i];
    }
}
```



Q14. Write a C++ Program to Implement Circular Queue.(using class and object)

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

int q[5];
int f=-1,r=-1;
 void enq(int data)
  {

     if(f==(r+1)%5)
     cout<<"\n q is full";
     else if(f==-1 && r==-1)
  {
  f=0;
  r=0;
  q[r]=data;
  }
   else
   {
     r=(r+1)%5;
     q[r]=data;
         }

  }
  void deq()
  {
  if(f==-1 && r==-1)
  cout<<"queue is empty";
  else if(f==r)
  {
  cout<<"\n deleted element "<<q[f];
  f=-1;
  r=-1;
   }
   else
   {
     cout<<"\n deleted element "<<q[f];
     f=(f+1)%5;
     }


  }
  void dis()
```

```
  {
  int i;
    if(f==-1 && r==-1)
  cout<<"\nqueue is empty";
  else if(f<=r)
  {
  cout<<"\n elements of q";
  for(i=f;i<=r;i++)
  cout<<"\t %d"<<q[i];
  }
  else
  {
  cout<<"\n enter the element";
   for(i=f;i<=5;i++)
  cout<<"\t "<<q[i];
  for(i=0;i<=r;i++)
  cout<<"\t "<<q[i];
  }
  }

void main()
{
  clrscr();
  enq(10);
  enq(20) ;
  enq(30);
  enq(40);
  enq(50);
  dis();
  deq();
  deq();
  deq();
  dis();
 getch();
  }
```

```
elements of q   %d10    %d20    %d30    %d40    %d50
deleted element 10
deleted element 20
deleted element 30
elements of q   %d40    %d50_
```