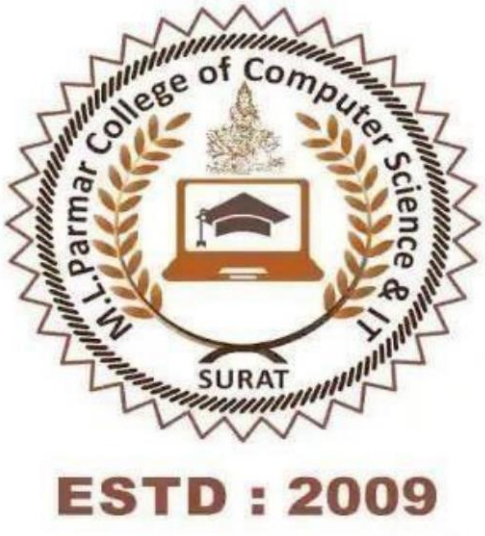
M.L.PARMAR COLLEGE
OF
COMPUTER SCIENCE & I.T.
OOPS and DS



Name	Vekariya Mohit Jaysukhbhai
Roll no	164
Div	2
Class	S.Y. BCA
SPID	2021051008

Assignment 2

Sr No.	Descreption	Page	Sign
12	Create a class STUDENT having data members as student number, student name. From the above class derive another class MARKS that contains data member as an array of five subject marks. Maximum marks of each subject are 100 marks. Derived a class RESULT from the above class MARKS having data members total, percentage and result (grade). Note that the passing criteria are 30 marks of each subject. Define a member function calculate () that calculates total, percentage and grade. The criteria for grade calculation: If percentage >=70% then grade A If percentage <70% and >=60% then grade B" If percentage <60% and >=50% then grade C" If percentage <50% and =30% then grade D" If percentage =30% then grade FAIL		
13	Write a program for creating a class figure and calculate the area of the circle (pi*r*r) with the help of constructor. Get the details from the user and display it.		
14	Create an abstract class “Shape” which stores data members like length, breath and radius. Create two classes “Circle” and “Rectangle “which stores data members like area respectively. Write a function to calculate area and display it.		
15	Build a class string. Use overloaded + to combine two string.		
16	Build a class string. Use overloaded == operator to compare two strings.		
17	Build a class string. Use overloaded operator = to copy one string into another		
18	Construct a class month with data members as month name. Write down overloaded operator ++ for the increment the month name by one. Write a member function read () and display () to read a name of month and print a name of month. Input:-		

	Enter name of a month: March Output:- Next month name is: April		
19	Construct a class month with data members as month name. Write down overloaded operator – for the previous month name. Write a member function read() and display () to read a name of a month and print a name of month. Input:- Enter name of a month: June Output:- Next month name is: May		
20	Write a program to do all operation of stack.		
21	Write a program to convert given infix expression into its equivalent postfix expression.		
22	Write a program to convert given infix expression into its equivalent prefix expression.		
23	Write a program to evaluate given postfix expression.		
24	Write a program to evaluate given prefix expression.		
25	Write a program for Tower of Hanoi.		
26	Write a program to do all operation of simple queue.		
27	Write a program to do all operation of circular queue.		
28	Write a program to do all operation of double ended queue.		

12] Create a class STUDENT having data members as student number, student name. From the above class derive another class MARKS that contains data member as an array of five subject marks. Maximum marks of each subject are 100 marks. Derived a class RESULT from the above class MARKS having data members total, percentage and result (grade).

Note that the passing criteria are 30 marks of each subject. Define a member function calculate () that calculates total, percentage and grade.

The criteria for grade calculation:

If percentage >=70% then grade A

If percentage <70% and >=60% then grade B"

If percentage <60% and >=50% then grade C"

If percentage <50% and =30% then grade D"

If percentage =30% then grade FAIL

Define a member function display () that prints the result of the N number of student in the following format:

MARKSHEET NUMBER NAME M1 M2 M3 M4 M5 TOTAL % RESULT

1. ... .

2. ... ...

3. ....

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
class Student
{
public:
   int st_number;
   char st_name[32];
};
class Marks : public Student
{
public:
   int *marks;
```

```
    int maximum_marks, maximum_subjects;
};
class Result : public Marks
{
    int total, check;
    float percentage;
    char *grade;

public:
    void getMarks();
    void calculate();
    void disp();
    void getDetails();
};
void Result::getMarks()
{
    int i;
    check = 1;
    maximum_marks = 100;
    maximum_subjects = 5;
    marks = new int[maximum_subjects];
    cout << "Enter the marks of 5 subjects:";
    for (i = 0; i < maximum_subjects; i++)
    {
        cin >> marks[i];
        if (marks[i] > 100 || marks[i] < 0)
            check = 0;
    }
}
void Result::calculate()
```

```
{
   int i;
   total = 0;
   if (check == 0)
      return;
   for (i = 0; i < maximum_subjects; i++)
   {
      total = total + marks[i];
   }
   percentage = (float)total / (maximum_subjects * maximum_marks);
   percentage *= 100;
   if (percentage > 100 || percentage < 0)
      grade = "Error";
   if (percentage >= 70)
      grade = "A";
   else if (percentage >= 60)
      grade = "B";
   else if (percentage >= 50)
      grade = "C";
   else if (percentage >= 30)
      grade = "D";
   else
      grade = "Fail";
}
void Result::disp()
{
   int i;
   cout << st_number << "\t" << st_name << "\t";
   if (check == 0)
   {
```

```
        for (i = 0; i < 8; i++)
            cout << "Error\t";
        return;
    }
    for (i = 0; i < maximum_subjects; i++)
        cout << marks[i] << "\t";
    cout << total << "\t" << setprecision(1) << percentage << "\t"
         << grade << endl;
}
void Result::getDetails()
{
    cout << "Enter student no : \n";
    cin >> st_number;
    cout << "Enter the student name : \n";
    cin >> st_name;
}
void format()
{
    cout << "Number\t"
         << "Name\t"
         << "M1\t"
         << "M2\t"
         << "M3\t"
         << "M4\t"
         << "M5\t"
         << "Total\t"
         << "Per\t"
         << "result\n";
}
void Details()
```

```
{
   int n,i;
   cout << "Enter the number of students : ";
   cin >> n;
   a = new Result[n];
   for (i = 0; i < n; i++)
   {
         a[i].getDetails();
         a[i].getMarks();
      a[i].calculate();
      clrscr();
   }
   format();
   for (i = 0; i < n; i++)
      a[i].disp();
}
void main()
{
   clrscr();
   Details();
   getch();
}
```

No	Name	M1	M2	M3	M4	M5	Total	Per.	Result
1	mohit	80	89	79	81	71	400	80	A
2	venish	79	61	76	75	60	352	70.4	A
3	kartik	21	30	35	29	33	148	29.6	FAIL
4	trupt	63	64	58	65	51	301	60.2	B
5	nikhil	70	78	80	70	75	373	74.6	B

13) Write a program for creating a class figure and calculate the area of the circle (pi*r*r) with the help of constructor. Get the details from the user and display it.

```
#include<iostream.h>
#include<conio.h>
class figure
{
        public:
                float r,total;
        figure()
        {
                cout<<"Enter the value of Circle Radius : ";
                cin>>r;
                float pi=3.14;
                total=pi*r*r;
        }
        void display()
        {
                cout<<"Value of pi is :3.14"<<endl;
                cout<<"Value of R is : "<<r<<endl;
                cout<<"Area of Circle is : "<< total;
        }
};
void main()
{
        clrscr();
        figure obj;
        obj.display();
        getch();
```

```
Enter the value of Circle Radius : 3
Value of pi is :3.14
Value of R is : 3
Area of Circle is : 28.26_
```

14)Create an abstract class “Shape” which stores data members like length, breath and radius. Create two classes “Circle” and “Rectangle “which stores data members like area respectively. Write a function to calculate area and display it.

```
#include<iostream.h>
#include<conio.h>
class shape
{
	public:
		float l,b,r;
		virtual void getdata()=0;
};
class circle : public shape
{
	public:
		float area;
		void getdata()
		{
			cout<<"Enter Radius of Circle : ";
			cin>>r;
		}
		void calculate()
		{
			float pi=3.14;
			area=pi*r*r;
			cout<<"Area of Circle is : "<<area<<endl;
		}
};
class ractangle : public shape
{
	public:
		float area;
```

```
        void getdata()
        {
                cout<<"Enter Length of Ractangle : ";
                cin>>l;
                cout<<"Enter Breath of Ractangle : ";
                cin>>b;
        }
        void calculate()
        {
                area=l*b;
                cout<<"Area of Ractangle is : "<<area;
        }
};
void main()
{
        clrscr();
        circle obj;
        ractangle obj2;
        obj.getdata();
        obj2.getdata();
        obj.calculate();
        obj2.calculate();
        getch();
}
```

```
Enter Radius of Circle : 2
Enter Length of Ractangle : 3
Enter Breath of Ractangle : 4
Area of Circle is : 12.56
Area of Ractangle is : 12
```

15)Build a class string. Use overloaded + to combine two string.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class string
{
        char string1[100];
        public:
        string()
        {
                strcpy(string1,"");
        }
        string(char s3[])
        {
                strcpy(string1,s3);
        }
        void display()
        {
                cout<<" String is : "<<string1<<endl;
        }
        string operator +(string str)
        {
                string t;
                strcpy(t.string1,string1);
                strcat(t.string1,str.string1);
                return t;
        }
};
void main()
{
```

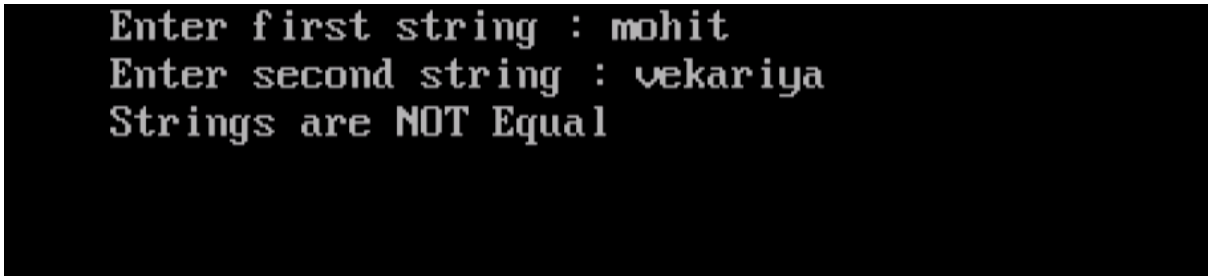
```
        clrscr();
        string s1="Vekariya";
        string s2="Mohit";
        string str;
        cout<<"First";
        s1.display();
        cout<<"Second";
        s2.display();
        str=s1+s2;
        cout<<"Concatination ";
        str.display();
        getch();
}
```

```
First String is : Vekariya
Second String is : Mohit
Concatination  String is : VekariyaMohit
_
```

16)Build a class string. Use overloaded == operator to compare two strings.

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
#include<conio.h>
class string
{
	char str[100];
	public:
		void getString()
		{
			gets(str);
		}
		int operator ==(string s)
		{
			if(strcmp(str,s.str))
			{
				return 0;
			}
			else
			{
				return 1;
			}
		}
};
void main()
{
	clrscr();
	string s1,s2;
	cout<<"Enter first string : ";
```

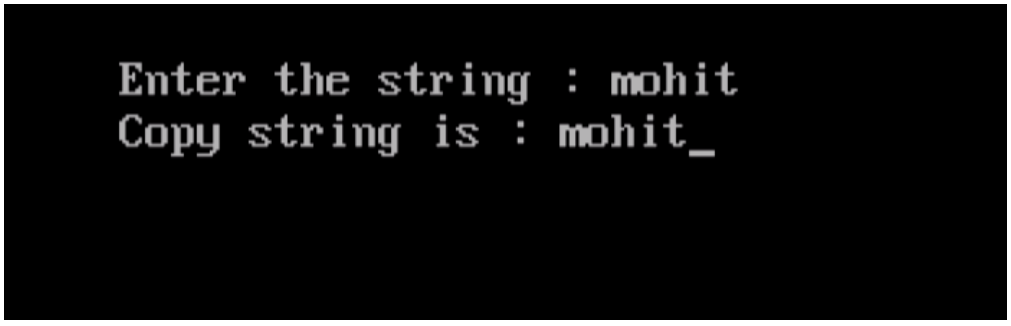
```
        s1.getString();
        cout<<"Enter second string : ";
        s2.getString();
        if(s1==s2)
        {
                cout<<"Strigs are Equal";
        }
        else
        {
                cout<<"Strings are NOT Equal";
        }
        getch();
}
```



17)Build a class string. Use overloaded operator = to copy one string into another.

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
class string
{       char str[30]; public:
        void getdata();
        void display();
        void operator=(string str1);
};
void string :: getdata()
{
```

```
        cout<<endl<<"Enter the string : "; cin>>str;
}
void string :: display()
{
        cout<<"Copy string is : "<<str;
}
void string :: operator=(string str1)
{
        strcpy(str1.str,str);
        cout<<"Copy string is : ";
}
void main()
{
        string s1;
        clrscr();
        s1.getdata();
        s1.display();
        getch();
}
```



18)Construct a class month with data members as month name. Write down overloaded operator ++ for the increment the month name by one. Write a member function read () and display () to read a name of month and print a name of month.

Input:-

Enter name of a month: March

Output:-

Next month name is: April

```
#include<iosstream.h>
#include<conio.h>
class month
{
        public:
                        char name[25];
                        int a;
                        month()
                        {
                                cout<<"Enter Month Name : ";
                                cin>>name;
                        }
                        void read()
                        {
                                        if (name == "january")
                                        {
                                                        a = 1;
                                        }
                                        if (name == "february")
                                        {
                                                        a = 2;
                                        }
                                        if (name == "march")
```

```
{
        a = 3;
}
if (name == "april")
{
        a = 4;
}
if (name == "may")
{
        a = 5;
}
if name =="june")
{
        a = 6;
}
if (name =="july")
{
        a = 7;
}
if (name =="august")
{
        a = 8;
}
if (name =="september")
{
        a = 9;
}
if (name =="october")
{
        a = 10;
```

```
                }
                if (name =="november")
                {
                        a = 11;
                }
                if (name =="december")
                {
                        a = 12;
                }
}
void display()
{
        a = a++;
        switch (a)
        {
                case 2:
                        cout << "next month is: February";
                        break;
                case 3:
                        cout << "next month is: March";
                        break;
                case 4:
                        cout << "next month is: April";
                        break;
                case 5:
                        cout << "next month is: May";
                        break;
                case 6:
                        cout << "next month is: June";
                        break;
```

```
                    case 7:
                        cout << "next month is: July";
                        break;
                    case 8:
                        cout << "next month is: August";
                        break;
                    case 9:
                        cout << "next month is: September";
                        break;
                    case 10:
                        cout << "next month is: October";
                        break;
                    case 11:
                        cout << "next month is: November";
                        break;
                    case 12:
                        cout << "next month is: December";
                        break;
                    case 13:
                        cout << "next month is: January";
                        break;
                    default:
                        cout << "invalid input";
                        break;
                }
            }
};
void main()
{
    clrscr();
```

```
        month m;
        m.read();
        m.display();
        getch();
}
```

```
Enter Month Name : january
next month is : February_
```

19)Construct a class month with data members as month name. Write down overloaded operator – for the previous month name. Write a member function read() and display () to read a name of a month and print a name of month.

Input:-

Enter name of a month: June

Output:-

Next month name is: May

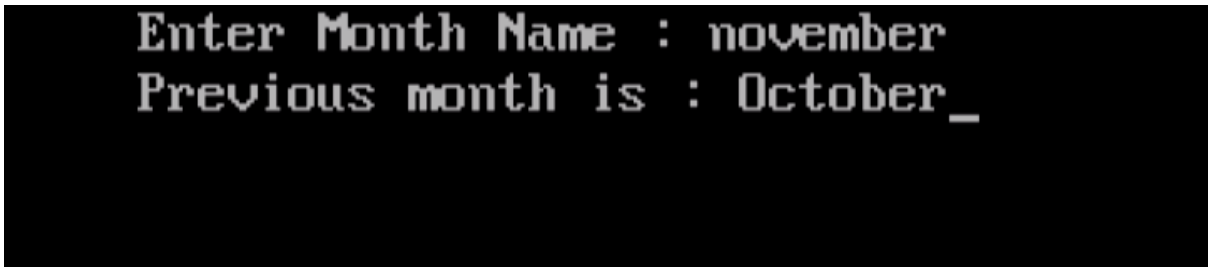
```
#include<iostream.h>
#include<conio.h>
class month
{
        public:
                        char name[25];
                        int a;
                        month()
                        {
                                cout<<"Enter Month Name : ";
                                cin>>name;
                        }
                        void read()
```

```
{
        if (name == "january")
        {
                a = 1;
        }
        if (name == "february")
        {
                a = 2;
        }
        if (name == "march")
        {
                a = 3;
        }
        if (name == "april")
        {
                a = 4;
        }
        if (name == "may")
        {
                a = 5;
        }
        if (name =="june")
        {
                a = 6;
        }
        if (name =="july")
        {
                a = 7;
        }
        if (name =="august")
```

```
                {
                        a = 8;
                }
                if (name =="september")
                {
                        a = 9;
                }
                if (name =="october")
                {
                        a = 10;
                }
                if (name =="november")
                {
                        a = 11;
                }
                if (name =="december")
                {
                        a = 12;
                }
}
void display()
{
        a = a-1;
        switch (a)
        {
                case 0:
                        cout << "Previous month is: December";
                        break;
                case 1:
                        cout << "Previous month is: January";
```

```
            break;
        case 2:
            cout << "Previous month is: Fabruary";
            break;
        case 3:
            cout << "Previous month is: March";
            break;
        case 4:
            cout << "Previous month is: April";
            break;
        case 5:
            cout << "Previous month is: May";
            break;
        case 6:
            cout << "Previous month is: June";
            break;
        case 7:
            cout << "Previous month is: July";
            break;
        case 8:
            cout << "Previous month is: August";
            break;
        case 9:
            cout << "Previous month is: September";
            break;
        case 10:
            cout << "Previous month is: Octber";
            break;
        case 11:
            cout << "Previous month is: November";
```

```
                                        break;
                                default:
                                        cout << "invalid input";
                                        break;
                                }
                        }
};
void main()
{
        clrscr();
        month m;
        m.read();
        m.display();
        getch();
}
```



20) Write a program to do all operation of stack

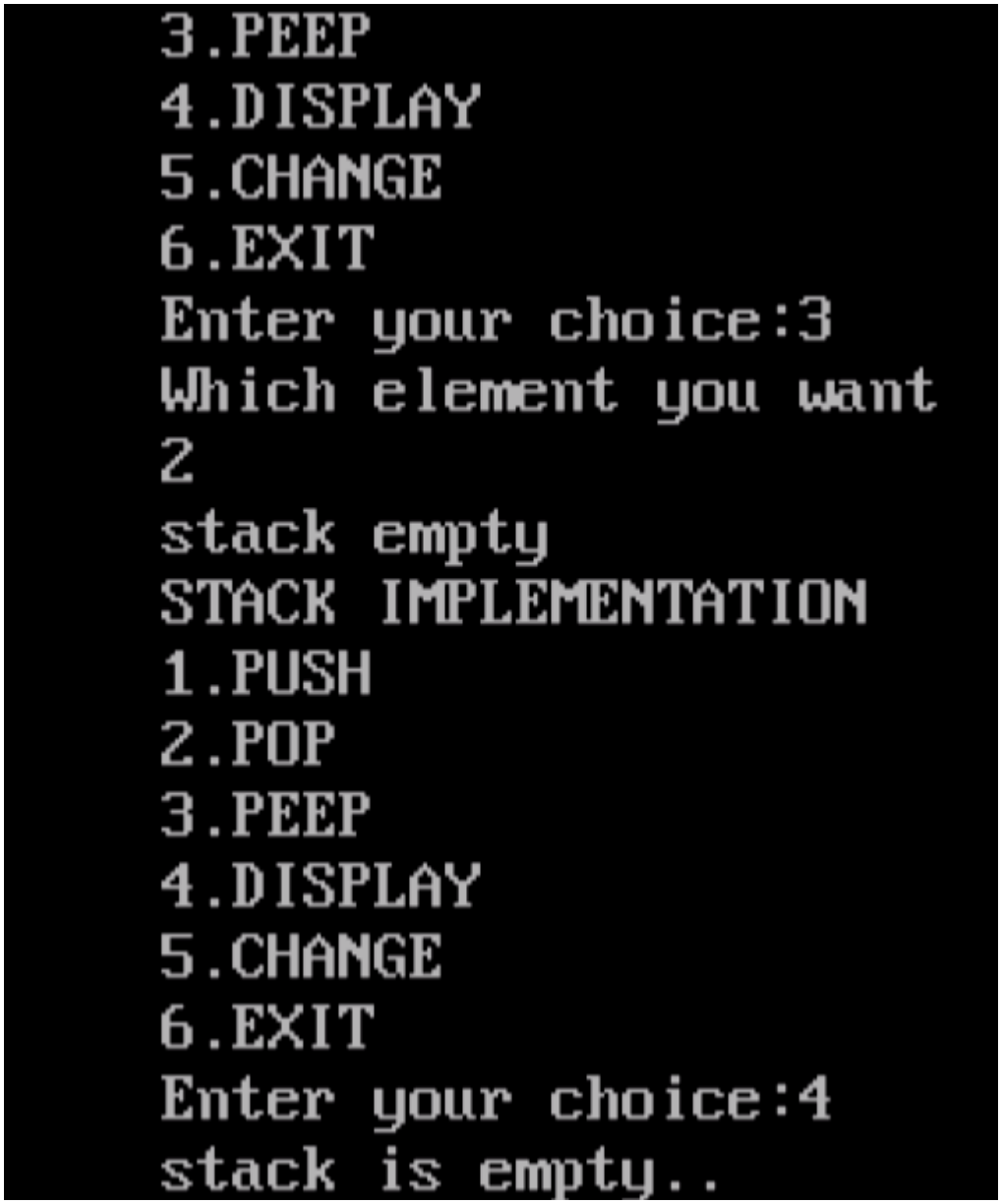
```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
class stack
{
        int n,top;
        int s[20];
        public:
        stack()
        {
                n=5;
                top=-1;
        }
        stack(int no)
        {
                n=no;
                top=-1;
        }
        void push(int data)
        {
                if(top==n-1)
                {
                        cout<<endl<<"stack overflow";
                }
                top++;
                s[top]=data;
        }
        void display()
```

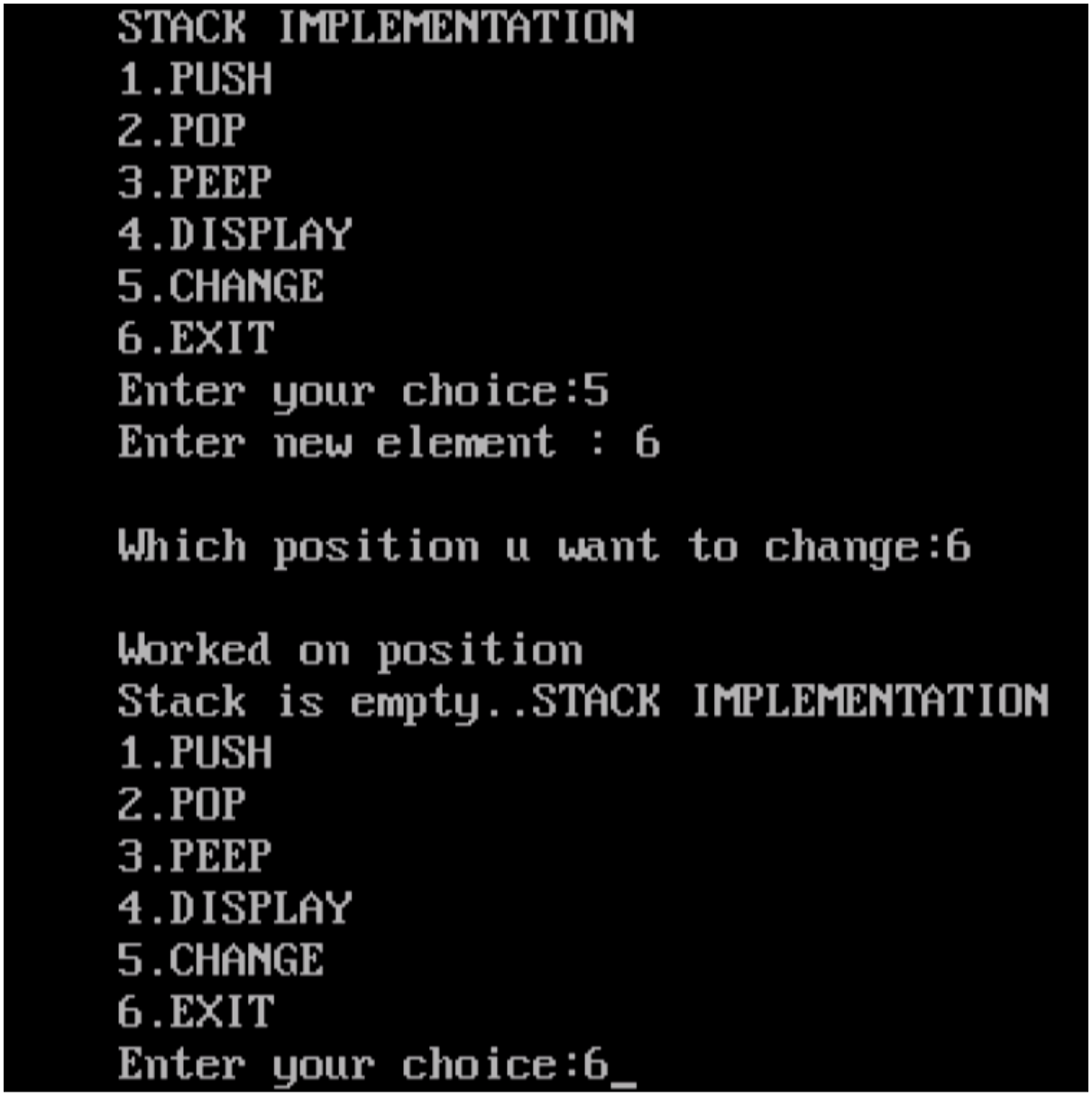
```
{
        if(top==-1)
                cout<<"stack is empty..\n";
        else
                cout<<endl<<"Content of stack::";
                for(int i=top;i>=0;i--)
                        cout<<"\n\t"<<s[i];
}
int pop()
{
        int data;
        if(top==-1)
        {
                cout<<"Stack is empty"<<endl;
                return 0;
        }
        data=s[top];
        top--;
        return data;
}
int peep()
{
        int i,data;
        cout<<"Which element you want"<<endl;
        cin>>i;
if((top-i+1)<0)
{
        cout<<"stack empty"<<endl;
        return 0;
}
```

```
            data=s[top-i+1];
            cout<<endl<<" TOP ========"<<endl<<top-i+1;
            return data;
        }
        void change(int data)
        {
            int i;
            cout<<"Which position u want to change:";
            cin>>i;
            cout<<endl<<"Worked on position";
        if((top-i+1)<0)
        {
            cout<<"Stack is empty..";
        }
            s[top-i+1]=data;
        }
};
void main()
{
        clrscr();
        int element,ch;
        stack s;
        do
        {
            cout<<endl<<"STACK IMPLEMENTATION"<<endl;
            cout<<"1.PUSH"<<endl;
            cout<<"2.POP"<<endl;
            cout<<"3.PEEP"<<endl;
            cout<<"4.DISPLAY"<<endl;
            cout<<"5.CHANGE"<<endl;
```

```
cout<<"6.EXIT"<<endl;
cout<<"Enter your choice:";
cin>>ch;
switch(ch)
{
        case 1:
                cout<<"Enter elemennt : ";
                cin>>element;
                s.push(element);
                break;
        case 2:
                element=s.pop();
                if(element!=0)
                        cout<<"the deleted element is : "<<element;
                break;
        case 3:
                element=s.peep();
                if(element!=0)
                        cout<<"selected element is : "<<element;
                break;
        case 4:
                s.display();
                break;
        case 5:
                cout<<"Enter new element : ";
                cin>>element;
                s.change(element);
                break;
        case 6:
                exit(0);
```

```
                default:
                        cout<<"wrong choice...";
                }
            }while(ch!=6);
        getch();
}
```


3.PEEP
4.DISPLAY
5.CHANGE
6.EXIT
Enter your choice:3
Which element you want
2
stack empty
STACK IMPLEMENTATION
1.PUSH
2.POP
3.PEEP
4.DISPLAY
5.CHANGE
6.EXIT
Enter your choice:4
stack is empty..



21) Write a program to convert given infix expression into its equivalent postfix expression.

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

```
class stack
{
        int n,top;
        char s[25],in[25],post[25];

        public:
                stack()
                {
                        n=5;
                        top=-1;
                }
                stack(int no)
                {
                        n=no;
                        top=-1;
                }
                void push(char data)
                {
                        top++;
                        s[top]=data;
                }
                char pop()
                {
                        char data;
                        data=s[top];
                        top--;
                        return data;
                }
                char peep()
                {
```

```
                char data;
                data=s[top];
                return data;
        }
};
void main()
{
        char in[25];
        stack s;
        clrscr();
        cout<<"Convert Infix Expression to Postfix Expression"<<endl;
        cout<<"Enter any infix expression : ";
        cin>>in;
        int l;
        l=strlen(in);
        s.push('(');
        in[l]=')';
        l++;
        in[l]='\0';
        cout<<"The postfix expression : ";
        for(int i=0;i<=l;i++)
        {
                switch(in[i])
                {
                        case'(':
                                s.push(in[i]);
                                break;

                        case'^':
                                if(s.peep()=='^')
```

```
            {
                cout<<s.pop();
            }
            s.push(in[i]);
            break;
        case'/':
            while(s.peep()=='^' || s.peep()=='/')
            {
                cout<<s.pop();
            }
            s.push(in[i]);
            break;
        case'*':
            while(s.peep()=='^' || s.peep()=='/'
            || s.peep()=='*')
            {
                cout<<s.pop();
            }
            s.push(in[i]);
            break;
        case'+':
            while(s.peep()=='^' || s.peep()=='/'
            || s.peep()=='*' || s.peep()=='+')
            {
                cout<<s.pop();
            }
            s.push(in[i]);
            break;
        case'-':
            while(s.peep()=='^' || s.peep()=='/'
```

```
                || s.peep()=='*' || s.peep()=='+'
                || s.peep()=='-')
                {
                        cout<<s.pop();
                }
                s.push(in[i]);
                break;
        case')':
                while(s.peep()!='(')
                {
                        cout<<s.pop();
                }
                s.pop();
                break;
        default:
                cout<<in[i];
        }
    }
    getch();
}
```

```
Convert Infix Expression to Postfix Expression
Enter any infix expression : a+(b*c-(d/e-f)*g)*h
The postfix expression : abc*de/f-g*-h*+
```

22) Write a program to convert given infix expression into its equivalent prefix expression.

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

```
class stack
{
        int n,top;
        char s[25];

        public:
                stack()
                {
                        n=10;
                        top=-1;
                }
                stack(int no)
                {
                        n=no;
                        top=-1;
                }
                void push(char data)
                {
                        top++;
                        s[top]=data;
                }
                char pop()
                {
                        char data;
                        data=s[top];
                        top--;
                        return data;
                }
                char peep()
                {
```

```
                char data;
                data=s[top];
                return data;
        }
};
void main()
{
        char in[25],pre[25];
        stack s;
        clrscr();

        cout<<"Convert Infix expression to Prefix expression"<<endl;
        cout<<"Enter the infix expression : ";
        cin>>in;

        int i,l,j=0;
        s.push('(');

        l=strlen(in);
        strrev(in);

        for(i=0;i<l;i++)
        {
                switch(in[i])
                {
                        case')':
                                s.push(in[i]);
                                break;
                        case'^':
                                while(s.peep()=='^')
```

```
                pre[j++]=s.pop();
            s.push(in[i]);
            break;
        case'/':
            while(s.peep()=='^'||s.peep()=='/')
            {
                pre[j++]=s.pop();
            }
            s.push(in[i]);
            break;
        case'*':
            while(s.peep()=='^'|| s.peep()=='/'||
            s.peep()=='*')

                pre[j++]=s.pop();
            s.push(in[i]);
            break;
        case'+':
            while(s.peep()=='^'|| s.peep()=='/'||
            s.peep()=='*'|| s.peep()=='+')

                pre[j++]=s.pop();
            s.push(in[i]);
            break;
        case'-':
            while(s.peep()=='^'|| s.peep()=='/'||
            s.peep()=='*'|| s.peep()=='+'|| s.peep()=='-')

                pre[j++]=s.pop();
```

```
                        s.push(in[i]);
                        break;
                case'(':
                        while(s.peep()!=')')

                                pre[j++]=s.pop();
                        s.pop();
                        break;
                default:
                        pre[j++]=in[i];
                }
        }
        while(s.peep()!='(')
                pre[j++]=s.pop();
        cout<<"The prefix expression : "<<strrev(pre);
        getch();
}
```

```
Convert Infix expression to Prefix expression
Enter the infix expression : a+(b*c-(d/e-f)*g)*h
The prefix expression : ¶Ò ¤¶¼!+a*-*bc*-/defgh_
```

23) Write a program to evaluate given postfix expression.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
```

```
class stack
{
        int s[10],top;

        public:
                stack()
                {
                        top=-1;
                }
                void push(int data)
                {
                        top++;
                        s[top]=data;
                }
                int pop()
                {
                        int data;
                        data=s[top];
                        top--;
                        return data;
                }
                void display()
                {
                        cout<<"Evaluation of Postfix Expression : "<<s[top];
                }
};
void main()
{
        char post[30],x[1];
        int l,a,b,rs,i;
```

```
stack s;
clrscr();

cout<<"Enter postfix expression : ";
cin>>post;

l=strlen(post);

for(i=0;i<l;i++)
{
        switch(post[i])
        {
                case'^':
                        a=s.pop();
                        b=s.pop();
                        rs=pow(b,a);
                        s.push(rs);
                        break;
                case'/':
                        a=s.pop();
                        b=s.pop();
                        rs=b/a;
                        s.push(rs);
                        break;
                case'*':
                        a=s.pop();
                        b=s.pop();
                        rs=b*a;
                        s.push(rs);
                        break;
```

```
                case'+':
                        a=s.pop();
                        b=s.pop();
                        rs=b+a;
                        s.push(rs);
                        break;
                case'-':
                        a=s.pop();
                        b=s.pop();
                        rs=b-a;
                        s.push(rs);
                        break;
                default:
                        x[0]=post[i];
                        x[1]='\0';
                        a=atoi(x);
                        s.push(a);
                        break;
            }
        }
        s.display();
        getch();
}
```

```
Enter postfix expression : 231*+9-
Evaluation of Postfix Expression : -4
```

24) Write a program to evaluate given prefix expression.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
class stack
{
        int s[10],top;

        public:
                stack()
                {
                        top=-1;
                }
                void push(int data)
                {
                        top++;
                        s[top]=data;
                }
                int pop()
                {
                        int data;
                        data=s[top];
                        top--;
                        return data;
                }
                void display()
                {
```

```
                cout<<"Evaluation of Prefix Expression : "<<s[top];
        }
};
void main()
{
        char pre[30],x[1];
        int l,a,b,rs,i;
        stack s;
        clrscr();

        cout<<"Enter Prefix expression : ";
        cin>>pre;

        l=strlen(pre);
        strrev(pre);

        for(i=0;i<l;i++)
        {
                switch(pre[i])
                {
                        case'^':
                                a=s.pop();
                                b=s.pop();
                                rs=pow(a,b);
                                s.push(rs);
                                break;
                        case'/':
                                a=s.pop();
                                b=s.pop();
                                rs=a/b;
```

```
                    s.push(rs);
                    break;
                case'*':
                    a=s.pop();
                    b=s.pop();
                    rs=a*b;
                    s.push(rs);
                    break;
                case'+':
                    a=s.pop();
                    b=s.pop();
                    rs=a+b;
                    s.push(rs);
                    break;
                case'-':
                    a=s.pop();
                    b=s.pop();
                    rs=a-b;
                    s.push(rs);
                    break;
                default:
                    x[0]=pre[i];
                    x[1]='\0';
                    a=atoi(x);
                    s.push(a);
                    break;
            }
        }
        s.display();
        getch();
```

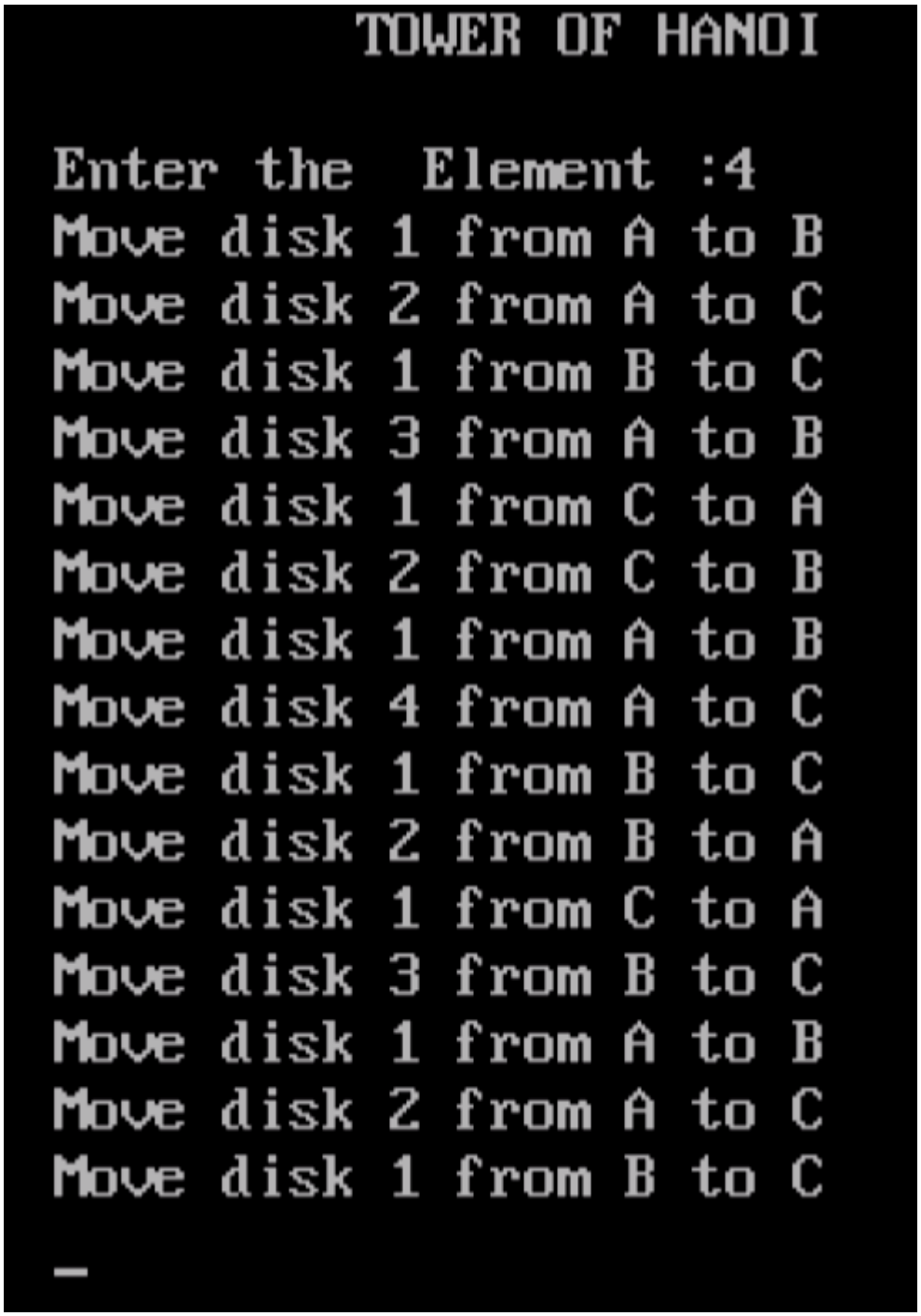
}

```
Enter Prefix expression : -*63-41
Evaluation of Prefix Expression : 15
```

25) Write a program for Tower of Hanoi.

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class tower
{
        int n;
        char a,b,c;
        public:
                tower()
                {

                }
                tower(int no,char t1,char t2,char t3)
                {
                        n=no;
                        a=t1;
                        b=t2;
                        c=t3;
                }
                void toh()
                {
                        if(n==1)
```

```
                {
                        cout<<"Move Disk "<<n<<" from tower "<<a<<" to
"<<c;
                        return;
                }
                tower(n-1,a,b,c);
                cout<<"Move Disk "<<n<<" from tower "<<a<<" to "<<c;
                tower(n-1,b,a,c);
        }
};
void main()
{
        int n;
        clrscr();
        cout<<"Tower Of Hanoi"<<endl;
        cout<<"enter the value of n : ";
        cin>>n;
        tower t(n,'A','B','C');
        t.tower();
        getch();
}
```



26) Write a program to do all operation of simple queue.

#include <iostream.h>

#include <conio.h>

#include <stdio.h>

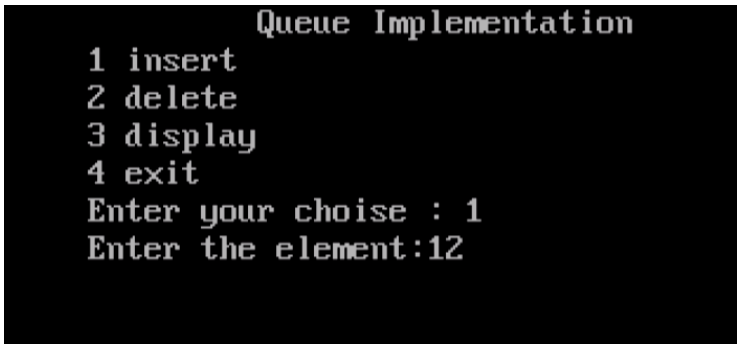
#include<stdlib.h>

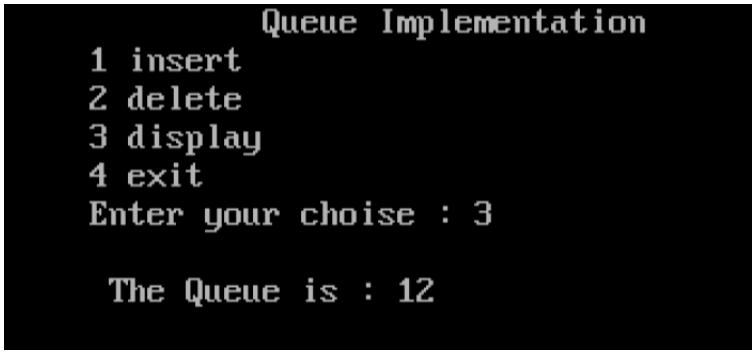
```
class queue
{
        int q[10], f, r, n;

        public:
                queue()
                {
                        f = -1;
                        r = -1;
                }
        void insert(int data)
        {
                if (r == n - 1)
                {
                        cout << "\n\t queue is overflow:";
                        return;
                }
                r++;
                q[r] = data;
                if (f == -1)
                        f = 0;
        }
        int del()
        {
                int data;
                if (f == -1)
                {
                        cout << "\n\t queue is underflow:";
                        return 0;
                }
```

```
                data = q[f];
                if (f == r) // q is empty?
                {
                        f = -1;
                        r = -1;
                }
                else
                f++;
                return data;
        }
        void display()
        {
                int i;
                if (f == -1)
                        cout << "\n\t Queue is empty:";
                else
                {
                        cout << "\n The Queue is : ";
                        for (i = f; i <= r; i++)
                        {
                                cout << q[i] << "\t";
                        }
                }
        }
};
void main()
{
   int e, ch;
   queue q;
   do
```

```
{
   clrscr();
      cout << "\t Queue Implementation\n";
      cout << "1 insert"<<endl;
      cout << "2 delete"<<endl;
      cout << "3 display"<<endl;
      cout << "4 exit"<<endl;
      cout << "Enter your choise : ";
      cin >> ch;
      switch (ch)
      {
      case 1:
         cout << "Enter the element:";
      cin >> e;
      q.insert(e);
      break;
   case 2:
      e = q.del();
      if (e != 0)
         cout << "\n\t The deleted element is" << e;
      break;
   case 3:
      q.display();
      break;
   case 4:
      cout << "\n Bye bye";
      exit(0);
      break;
   default:
      cout << "wrong choise";
```

```
    }
    getch();
  } while (ch != 4);
}
```





27) Write a program to do all operation of circular queue.

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
class queue
{
  int q[10], f, r, n;

public:
  queue()
  {
```

```
    f = -1;
    r = -1;
    n = 10;
}
void insert(int data)
{
    if ((f == 0 && r == n - 1) || (f - r == 1))
    {
        cout <<"Circular Queue is Overflow";
        return;
    }
    if (r == n - 1)
        r = 0;
    else
        r++;
    q[r] = data;
    if (f == -1)
        f = 0;
}
int del()
{
    int data;
    if (f == -1)
    {
        cout <<"Circular Queue is Underflow";
        return 0;
    }
    data = q[f];
    if (f == r)
    {
```

```
            f = -1;
            r = -1;
        }
        else if (f == n - 1)
            f = 0;
    else
        f++;
    return data;
}
void display()
{
    int i;
    if (f == -1)
            cout << "Circular Queue is empty"<<endl;
        else
        {
            cout << "The circular queue is : ";
            if (f > r)
        {
            for (i = f; i <= n - 1; i++)
                    cout << q[i] <<endl;
                for (i = 0; i <= r; i++)
                    cout << q[i] <<endl;
        }
        else
        {
            for (i = f; i <= r; i++)
                    cout << q[i] <<endl;
        }
    }
```

```
    }
};
void main()
{
    int e, ch;
    queue q;
    clrscr();
    do
    {
            cout << "\t\tCircular Queue Implementation"<<endl;
            cout << "1 Insert"<<endl;
            cout << "2 Delete"<<endl;
            cout << "3 Display"<<endl;
            cout << "4 Exit"<<endl;
            cout << "Enter your choise : ";
        cin >> ch;
        switch (ch)
        {
        case 1:
                cout << "Enter the element of data : ";
            cin >> e;
            q.insert(e);
            break;
        case 2:
            e = q.del();
            if (e != 0)
                    cout << "The Deleted element is : "<<e<<endl;
            break;
        case 3:
            q.display();
```

```
        break;
      case 4:
        cout << "\n Bye bye";
        break;
      default:
          cout << "=*=*=*=*=*=*= Wrong Choise =*=*=*=*=*=*=";
        }
   } while (ch != 4);
   getch();
}
```

```
    4 Exit
    Enter your choise : 1
    Enter the element of data : 12
                    Circular Queue Implementation
    1 Insert
    2 Delete
    3 Display
    4 Exit
    Enter your choise : 2
    The Deleted element is : 12
                    Circular Queue Implementation
    1 Insert
    2 Delete
    3 Display
    4 Exit
    Enter your choise : 3
    Circular Queue is empty
                    Circular Queue Implementation
    1 Insert
    2 Delete
    3 Display
    4 Exit
    Enter your choise : 4

     Bye bye_
```

28) Write a program to do all operation of double ended queue.

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
class dqueue
{
   int q[10], f, r, n;

public:
   dqueue()
   {
      f = -1;
      r = -1;
      n = 10;
   }
   void dqinsert_left(int data)
   {
      if (f == 0)
      {
            cout << "Dqueue full form left side : ";
         return;
      }
      if (r == -1)
      {
         f = n - 1;
         r = n - 1;
      }
      else
         f--;
      q[f] = data;
```

```
}
int dqinsert_right(int data)
{
   if (r == n - 1)
   {
         cout << "Dqueue full from right side : ";
      return 0;
   }
   r++;
   q[r] = data;
   if (f == -1)
      f = 0;
}
int dqdelete_left()
{
   int data;
   if (f == -1)
   {
         cout << "Dq is Underflow";
      return 0;
   }
   data = q[f];
   q[f] = 0;
   f++;
   if (f > r)
   {
      f = -1;
      r = -1;
   }
   return data;
```

```
}
int dqdelete_right()
{
   int data;
   if (r == -1)
   {
         cout << "Dq is Underflow";
      return 0;
   }
   data = q[r];
   q[r] = 0;
   r--;
   if (f > r)
   {
      f = -1;
      r = -1;
   }
   return 0;
}
void display()
{
   int i;
   if (f == -1)
         cout << "Dq is empty";
   else
   {
      if (f > r)
      {
         for (i = f; i < n - 1; i++)
            cout << q[i] << "\t";
```

```
            for (i = 0; i <= r; i++)
                cout << q[i] << "\t";
        }
        else
        {
            for (i = f; i <= r; i++)
                cout << q[i] << "\t";
        }
      }
   }
};
void main()
{
   int e, ch;
   dqueue dq;
   clrscr();
   do
   {
         cout << "1 insert from left"<<endl;
         cout << "2 insert from right"<<endl;
         cout << "3 delete from left"<<endl;
         cout << "4 delete from right"<<endl;
         cout << "5 display"<<endl;
         cout << "6 Exit"<<endl;
         cout << "Enter your choise : ";
      cin >> ch;
      switch (ch)
      {
      case 1:
            cout << "Enter the element : ";
```

```
            cin >> e;
            dq.dqinsert_left(e);
            break;
        case 2:
                cout << "Enter The element : ";
            cin >> e;
            dq.dqinsert_right(e);
            break;
        case 3:
            e = dq.dqdelete_left();
            if (e != 0)
                    cout << "The deleted elemnt is : " << e;
            break;
        case 4:
            e = dq.dqdelete_right();
            if (e != 0)
                    cout << "The deleted elemnt is : " << e;
                break;
        case 5:
            dq.display();
            break;
        case 6:
                cout << "bye bye.....";
            break;
        default:
            cout << "wrong choise";
        }
        getch();
    } while (ch != 6);
}
```

```
1 insert from left
2 insert from right
3 delete from left
4 delete from right
5 display
6 Exit
Enter your choise : 1
Enter the element : 12
1 insert from left
2 insert from right
3 delete from left
4 delete from right
5 display
6 Exit
Enter your choise : 2
Enter The element : 13
Dqueue full from right side : 1 insert from left
2 insert from right
3 delete from left
4 delete from right
5 display
6 Exit
Enter your choise : 5
12
```