NAME: RABIA BATOOL

ROLL NO. : 2022-BSE-064

LAB#1

SUBJECT : DATA STRUCTURE AND ALGORITHIMS

SUBMITTED TO: SIR REHAN

Lab#1:

Write the prototype of a function named DS() which accepts an array of integers, a

pointer to double, a float by reference and returns a character.

Answer:

Char ds(int arr[],double *ptr,float &ref)

2 Write statements as directed:

int a[]={1,2,3,4,5};

int *p;

p = a;

//Print the elements of array using the mentioned notation:

for (int i = 0; i < 5; i++)

{

//Subscript notation with name of array

Cout<<”elements”<<a[i]<<endl;

//Subscript notation with pointer 'p'

Cout<<”elements”<<p[i]<<endl;

//Offset notation using array name

Cout<<elements using off set of array name”<<*(a+i)<<endl;

//Offset notation using pointerp

Cout<<elements using off set of pointer name”<<*(p+i)<<endl;

3. What is the difference between function overloading and function overriding?

Function overloading: refers to the ability to define **multiple functions** in the same class or scope with the same name **but different parameter lists** (i.e., a different number or type of parameters). It allows you to create functions with the same name that perform different tasks based on their parameter signatures.

Function overriding: is a feature of inheritance in object-oriented programming. It allows a subclass to provide **a specific implementation** of a function that is already defined in its superclass. Both the base class (superclass) and the derived class (subclass) have functions with the same name, same parameters, and the same return type.

4. Write C++ statement(s) to allocate space for 10 doubles (using dynamic memory allocation).

Double *ptr;

```
Ptr=new double[10];
```

5 Study the given program and determine what the program is intended to do. (Hint: Dry

run the program with array {1,2,1,2,3} and analyze the output).

Consider we have entered 1,2,1,2,3 in array.first the program will compare the first index value with the second index value if both are save it will remover the dublicate element and print the array unique elements so, the main purpose of program is to remove **dublicate elements** of array.

Write a C++ function which accepts an array of integers and the size of the array and finds :

a. Sum of the elements in the array

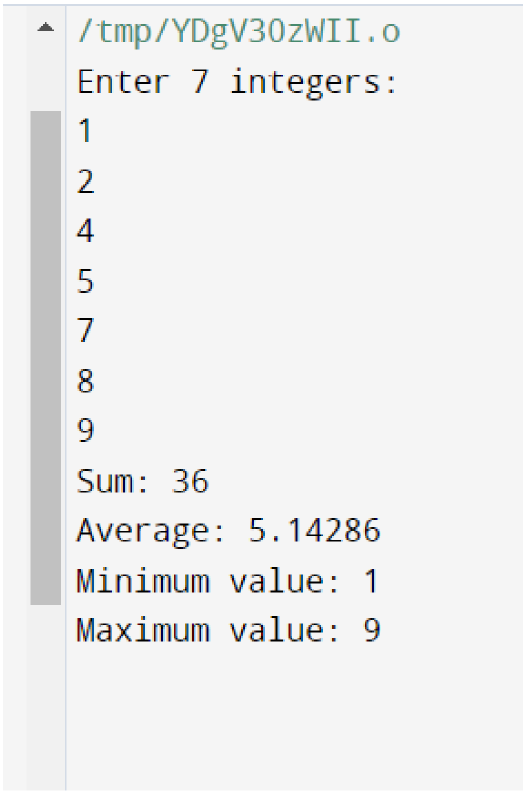
b. Average of the array elements

c. Minimum and maximum values in the array

In the main program, declare an array of 7 integers using dynamic memory allocation and call

the aforementioned function. Display the output of the function within the main.

```
include <iostream>

using namespace std;

void Array(int arr[], int size, int& sum, double& average, int& minVal, int& maxVal) {

    if (size <= 0) {

        cout << "Array is empty. Cannot calculate statistics." << endl;

        return;

    }

    sum = 0;

    minVal = arr[0];

    maxVal = arr[0];


    for (int i = 0; i < size; i++) {

        sum += arr[i];

        if (arr[i] < minVal) {
            minVal = arr[i];
        }
        if (arr[i] > maxVal) {
            maxVal = arr[i];
        }
    }
    average = (sum) / size;
}

int main() {
    int size = 7;
    int* arr = new int[size];
    std::cout << "Enter " << size << " integers:" << std::endl;
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }
    int sum, minVal, maxVal;
    double average;
    findArrayStats(arr, size, sum, average, minVal, maxVal);
    cout << "Sum: " << sum << endl;
    cout << "Average: " << average << endl;
    :cout << "Minimum value: " << minVal << endl;
    :cout << "Maximum value: " << maxVal << endl;
    delete[] arr;

    return 0;
}
```



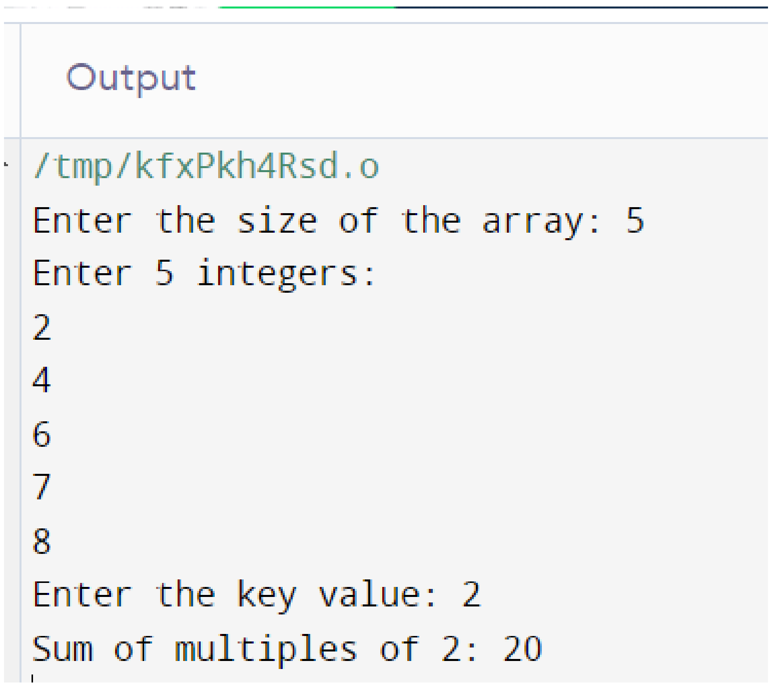
```
/tmp/YDgV30zWII.o
Enter 7 integers:
1
2
4
5
7
8
9
Sum: 36
Average: 5.14286
Minimum value: 1
Maximum value: 9
```

Exercise 2

Write a program with a function which accepts an array of integers and a key value. The function should return the sum of all the multiples of the key value in the array. For example, for the array {1, 4, 10, 12, 15, 20, 22} and the key value 5, the function should return the sum 10+15+20.

Task#2

```
#include <iostream>

int sumOfMultiples(int arr[], int size, int key) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        if (arr[i] % key == 0) {
            sum += arr[i];
        }
    }
    return sum;
int main() {
    int size;    std::cout << "Enter the size of the array: ";
    std::cin >> size;
    int* arr = new int[size]
    scout << "Enter " << size << " integers:" << endl;
    for (int i = 0; i < size; i++) {
scin >> arr[i];
    }
 int key;
    std::cout << "Enter the key value: ";
    std::cin >> key;
    int result = sumOfMultiples(arr, size, key);
    std::cout << "Sum of multiples of " << key << ": " << result << std::endl;
    delete[] arr;

    return 0;
}
```


Output

```
/tmp/kfxPkh4Rsd.o
Enter the size of the array: 5
Enter 5 integers:
2
4
6
7
8
Enter the key value: 2
Sum of multiples of 2: 20
```

Task #3

Exercise 3

Create a class Matrix to model 2x2 matrices. Provide a default , parameterized

constructor,get_input method to assign values to the matrix. Using a member function,

Lab 1

Data Structures and Algorithms Page 4

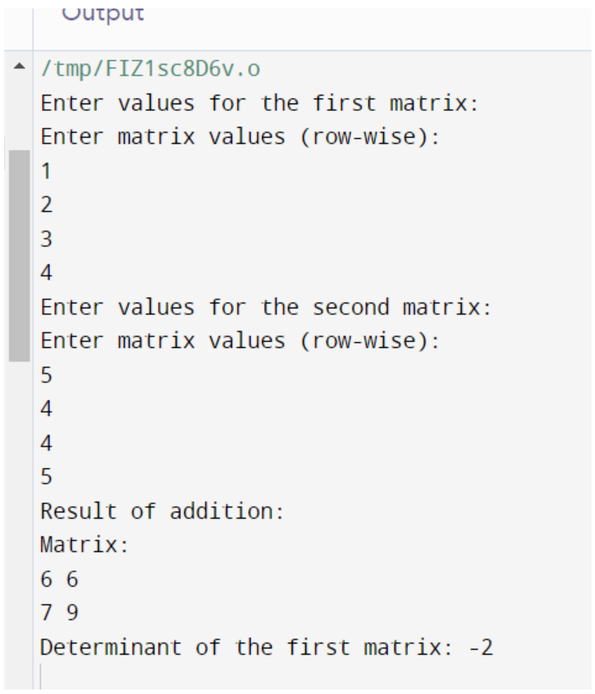
overload the '+' operator to add two matrices. Likewise, overload the '~' operator to find the

determinant of the matrix. Also provide a display() member function to print the matrix. In the

main program, create an object of class Matrix and call its member functions.

```
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;


class Matrix {
```

```
private:
    int mat[2][2];
public:
    Matrix() {
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                mat[i][j] = 0;
            }
        }
    }
    Matrix(int a, int b, int c, int d) {
        mat[0][0] = a;
        mat[0][1] = b;
        mat[1][0] = c;
        mat[1][1] = d;
    }
    void get_input() {
        std::cout << "Enter matrix values (row-wise):" << std::endl;
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                std::cin >> mat[i][j];
            }
        }
    }
  Matrix operator+(const Matrix& other) {
        Matrix result;
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                result.mat[i][j] = mat[i][j] + other.mat[i][j];
            }
        }
        return result;
    }
    int operator~() {
        return (mat[0][0] * mat[1][1] - mat[0][1] * mat[1][0]);
    }

    void display() {
cout << "Matrix:" << endl;
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                cout << mat[i][j] << " ";
            }
cout << std::endl;
       };

int main() {
    Matrix matrix1, matrix2, result;
    std::cout << "Enter values for the first matrix:" << std::endl;
    matrix1.get_input();
   std::cout << "Enter values for the second matrix:" << endl;
    matrix2.get_input();
    result = matrix1 + matrix2;
    std::cout << "Result of addition:" << std::endl;
    result.display();
    int determinant = ~matrix1;
    std::cout << "Determinant of the first matrix: " << determinant << std::endl;
    return 0;

}
```



Output

```
/tmp/FIZ1sc8D6v.o
Enter values for the first matrix:
Enter matrix values (row-wise):
1
2
3
4
Enter values for the second matrix:
Enter matrix values (row-wise):
5
4
4
5
Result of addition:
Matrix:
6 6
7 9
Determinant of the first matrix: -2
```

Fatima Jinnah Women University

Data Structure & Algorithm

LAB#01

Task 1 :

Give answers to the following.

1. Write the output of the following program.

```
#include<iostream>
using namespace std;
int mystery(int,int);
int main()
{
      int x=5,y=2;
      cout<<"Result = "<<mystery(x,y);
      return 0;
}
int mystery(int a, int b)
{
      if(b==1)
            return a;
      else
            return a + mystery(a, b-1);
}
```

Answer:

10

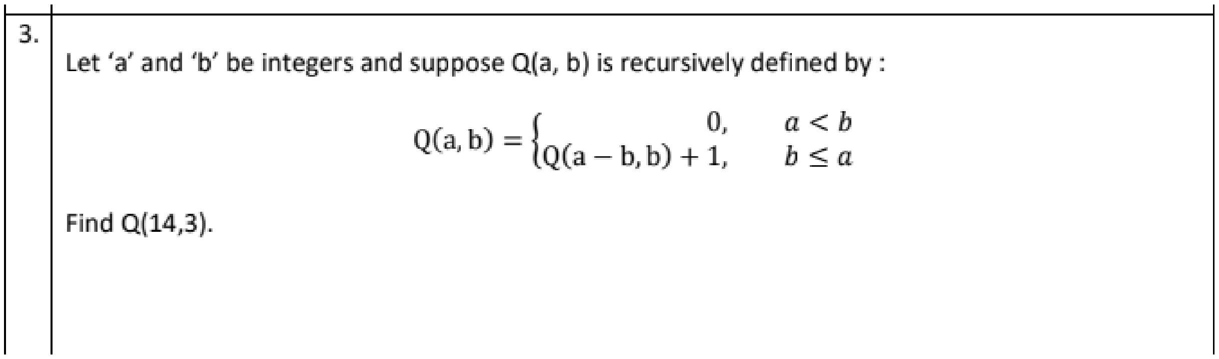
2. Let J and K be integers and suppose Q(J, K) is recursively defined by :

$$Q(J,K) = \begin{cases} 5, & J < K \\ Q(J-K, K+2) + J, & J \geq K \end{cases}$$

Trace and Find Q(5, 3).

Answer:

10



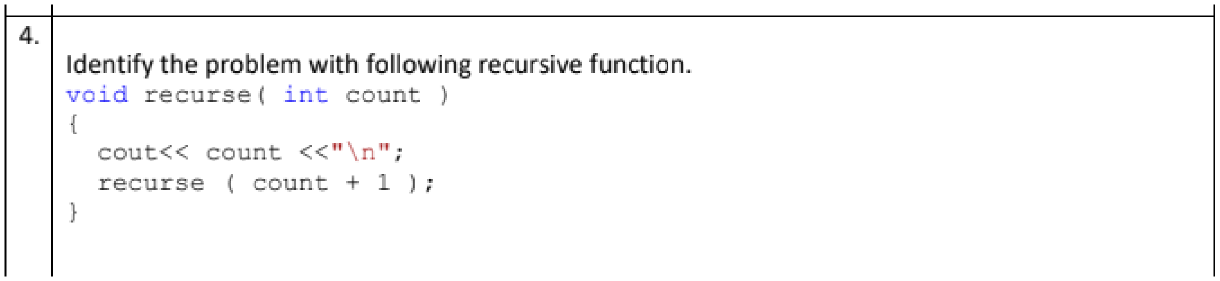
3. Let ‘a’ and ‘b’ be integers and suppose Q(a, b) is recursively defined by :

$$Q(a,b) = \begin{cases} 0, & a < b \\ Q(a-b,b)+1, & b \le a \end{cases}$$

Find Q(14,3).

Answer:

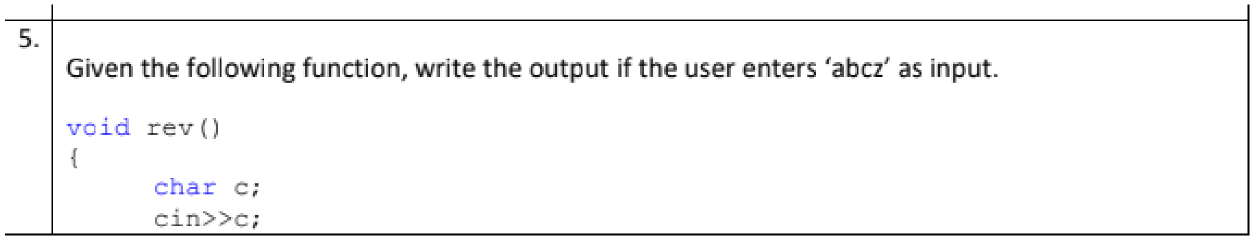
4



4. Identify the problem with following recursive function.

```
void recurse( int count )
{
  cout<< count <<"\n";
  recurse ( count + 1 );
}
```

In this function no condition is provided.It will be executed repeatedly for indefinite time. To fix this function, you need to add a base case that specifies when the recursion should terminate. For example, you could set a condition where the recursion stops when count reaches a certain value.



5. Given the following function, write the output if the user enters ‘abcz’ as input.

```
void rev()
{
      char c;
      cin>>c;
```

```
      if(c!='z'){
            rev();
            cout<<c;
      }
}
```

Answer:

cba

Task 2 :

Implement the following exercises.

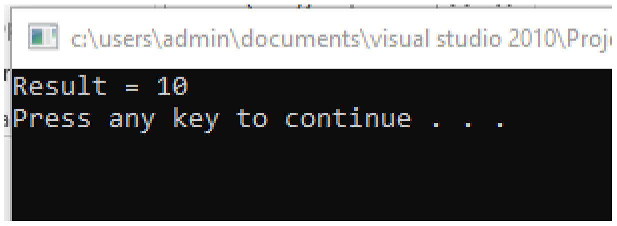
Exercise 1

Write a function sum(int a[], int size) to (recursively) compute the sum of the elements in an array.

Example Run :

```
int arr[]={1,2,3,4} ;
int result = sum(arr,4) ;
cout<<result<<endl ; //Should print 10
```

```
// 147.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;

int sum(int arr[] , int a);




int _tmain(int argc, _TCHAR* argv[])
{
int arr[]={1,2,3,4};
cout<<"Result = "<<sum(arr,4)<<endl;
system("pause");
return 0;

}
int sum(int arr[], int a)
{
if(a==0)
return 0;
else
return arr[a-1]+sum(arr,a-1);
}
```



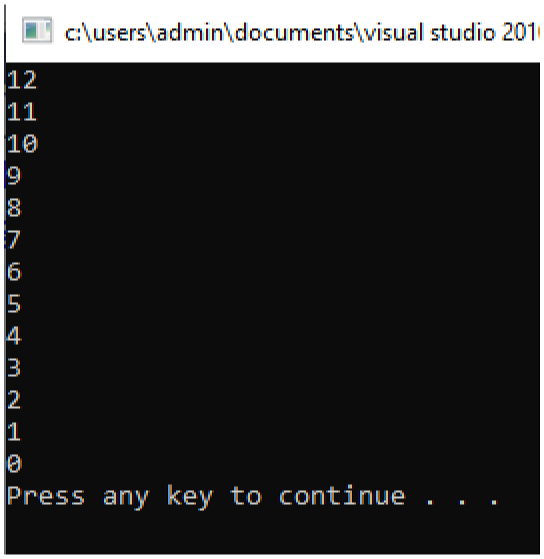
c:\users\admin\documents\visual studio 2010\Proj

```
Result = 10
Press any key to continue . . .
```

Exercise 2

Write a recursive function to print integers from a given number N to 0. When called as `print (10)`, the function should print : 10 9 8 7 6 5 4 3 2 1 0

```
// 147.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
void print( int a);




int _tmain(int argc, _TCHAR* argv[])
{
int a=10;
print(a);
system("pause");
return 0;

}
void print( int a)
{
if(a!=-1)
{cout<<a<<endl;
print (a-1);}

}
```



```
12
11
10
9
8
7
6
5
4
3
2
1
0
Press any key to continue . . .
```

Exercise 3

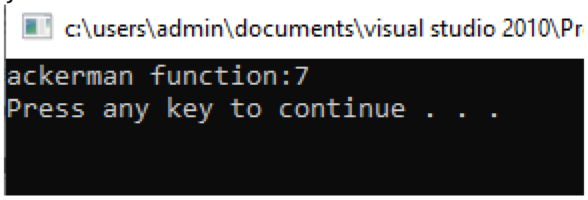
Ackermann's function is defined recursively on non-negative integers as follows.

```
A(m,n) = n+1                      if m == 0
A(m,n) = A(m-1, 1)                if m != 0, n == 0
A(m,n) = A(m-1, A(m, n-1))        if m != 0, n != 0
```

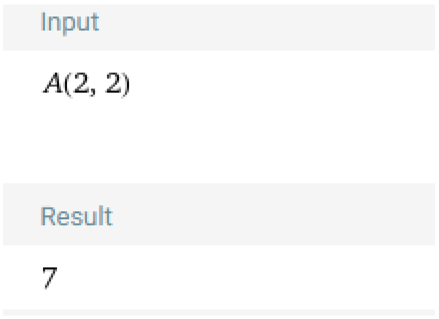
Implement it as a recursive function Ackermann(M,N) which takes two positive integers as input and returns a positive integer as result. Once implemented test your program by evaluating Ackermann(2,2).

```
// 147.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
int ackerman(int m,int n);
int _tmain(int argc, _TCHAR* argv[])
{int a=2,b=2;
cout<<"ackerman function:"<<ackerman( a, b)<<endl;
system("pause");
return 0;

}
int ackerman(int m,int n)
{
if(m==0)
return n+1;
if ((m!= 0 )&&( n== 0))
```

```
        return ackerman(m-1, 1);
if(( m!= 0) &&( n!= 0))
        return ackerman(m-1, ackerman(m, n-1));
}
```



```
c:\users\admin\documents\visual studio 2010\Pr
ackerman function:7
Press any key to continue . . .
```



Input

$A(2, 2)$

Result

7

Exercise 4

Binomial coefficients are normally computed using the following formula.

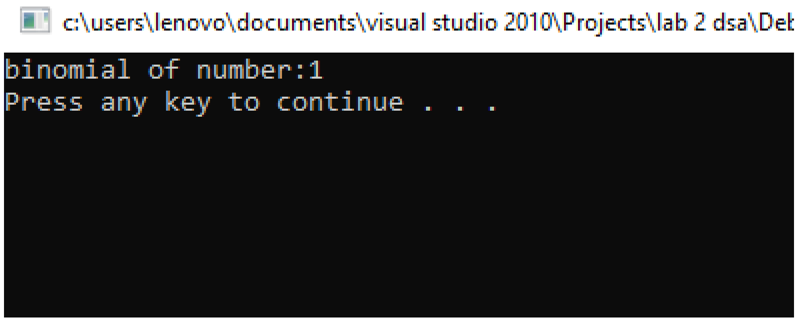
$$\binom{n}{m} = \frac{n!}{(n-m)!m!}$$

Binomial coefficients can also be computed using the following recursive definition.

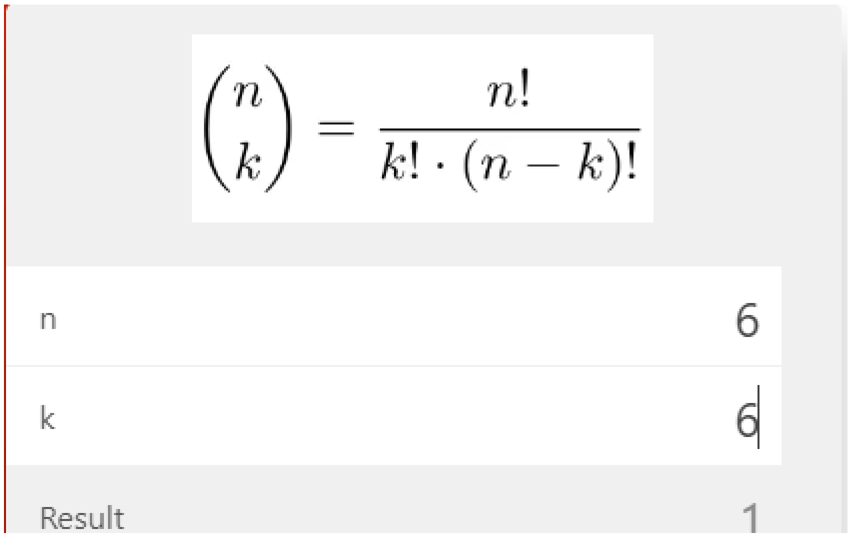
$$\binom{n}{m} = \begin{cases} 1 & m = 0, \\ 1 & n = m, \\ \binom{n-1}{m} + \binom{n-1}{m-1} & \text{otherwise.} \end{cases}$$

Write a C++ program to compute binomial coefficients using the mentioned recursive definition.

```
// 147.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
int bi(int m,int n);
int _tmain(int argc, _TCHAR* argv[])
{int a=5,b=5;
cout<<"ackerman function:"<<bi( a, b)<<endl;
system("pause");
return 0;
}
int bi(int m,int n)
{

if ((m== 0 )||( n==m ))
        return  1;
else
        return (bi(n-1),m)+bi((n-1),(m-1));
}
```



```
c:\users\lenovo\documents\visual studio 2010\Projects\lab 2 dsa\Deb
binomial of number:1
Press any key to continue . . .
```



$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

n	6
k	6
Result	1

Name :Rabia batool

Roll no:2022-BSE-052

LAB 3

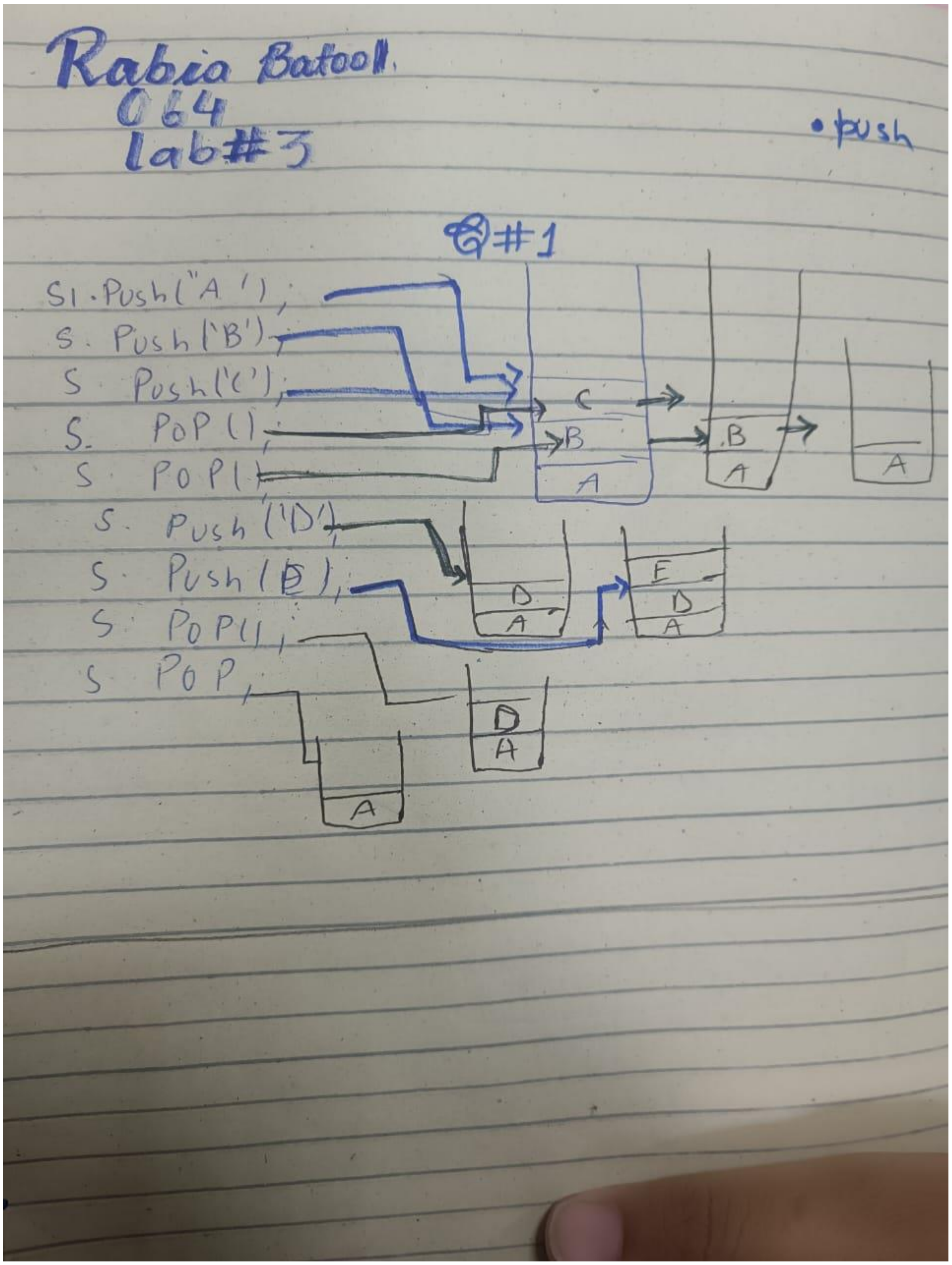
Submitted to :

Sir Rehan

TASK – 01

CODE TASK 1

Implement the Stack class and its basic functions and test all the functions in main constructor, push(), pop(), Is_empty(), Is_full(), Display(), Top()



Rabia Batool.
064
lab#3

• push

Q#1

```
S1.Push("A');
S. Push('B');
S  Push('C');
S.  PoP();
S  PoP();
S. Push('D');
S. Push(E);
S. PoP();
S  PoP;
```

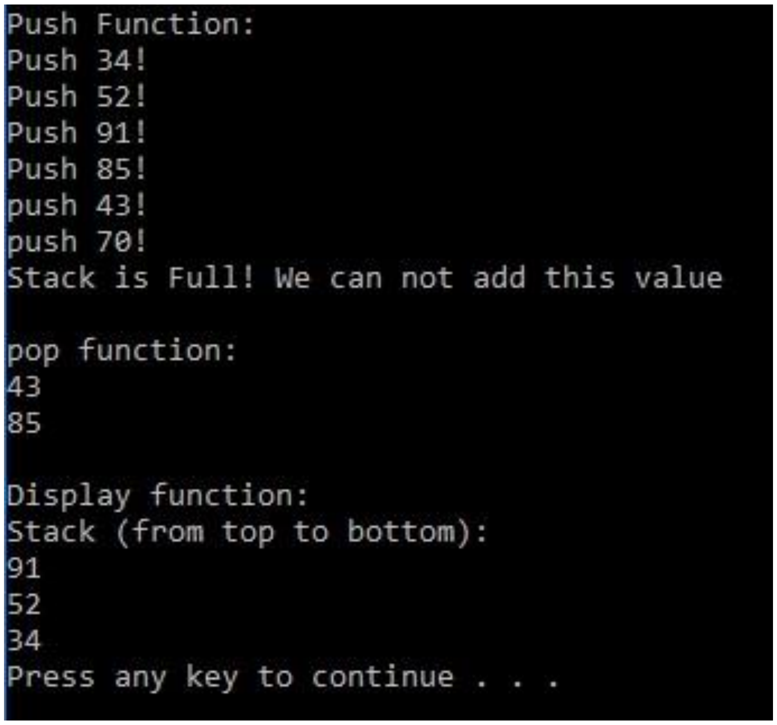
CODE:

```
#include"stdafx.h"

#include<iostream> using

namespace std;


const int size=5;


class Stack{

            private:

                        int array[size];

                        int top;

            public:

               Stack(){

                        top=-1;

                        }


                        void push(int val){

                                  if(top<size-1){

                                            ++top;

                                            array[top]=val;

                                  }

                                  else

                                     cout<<"Stack is Full! We can not add this value\n";

                        }


                        int pop(){

                                  if(top>=0){

                        return array[top--];

                                  }
```

```
                else{
                        cout<<"Stack is empty!\nWe are displaying garbage values...\n";
                }
        }

         bool empty(){
                if(top<0){
    return true;
                }
                else
                 return false;
        }

        bool full(){
if(top=size-1){
                        return true;
                }
                else
                 return false;
        }

         void display(){
                 if(!empty()){
                          cout<<"Stack (from top to bottom):\n";
                         for(int i=top;i>=0;i--){
                                cout<<array[i]<<endl;
                        }
                }
                else
```

```
                                        cout<<"Stack is empty!\n";
                        }

};

int main(){
            Stack s1;
   cout<<"Push Function:\n";     cout<<"Push
34!\n";
            s1.push(34);
                cout<<"Push 52!\n";
            s1.push(52);
                cout<<"Push 91!\n";
           s1.push(91);      cout<<"Push
85!\n";             s1.push(85);
cout<<"push 43!\n";      s1.push(43);
             cout<<"push 70!\n";
            s1.push(70); cout<<"\npop function:
            \n"; cout<<s1.pop()<<endl;
            cout<<s1.pop()<<endl;
            cout<<"\nDisplay function: \n";
            s1.display();


            system("pause");
            return 0;
}
```

<u>OUTPUT:</u>



```
Push Function:
Push 34!
Push 52!
Push 91!
Push 85!
push 43!
push 70!
Stack is Full! We can not add this value

pop function:
43
85

Display function:
Stack (from top to bottom):
91
52
34
Press any key to continue . . .
```

CODE TASK 2

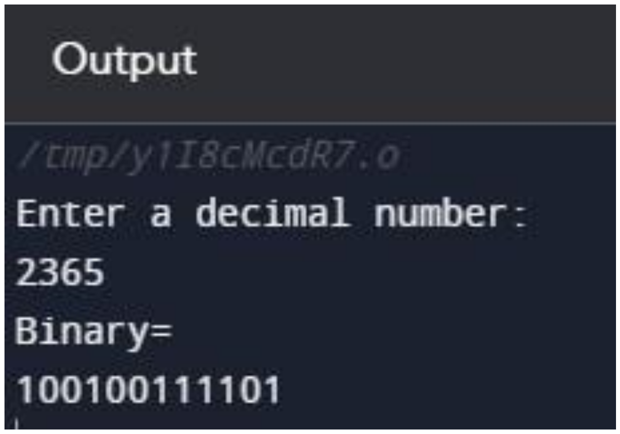
Write a C++ program that prompts user to enter a number (in decimal). Convert the number into binary and display the binary number using the Stack

CODE:

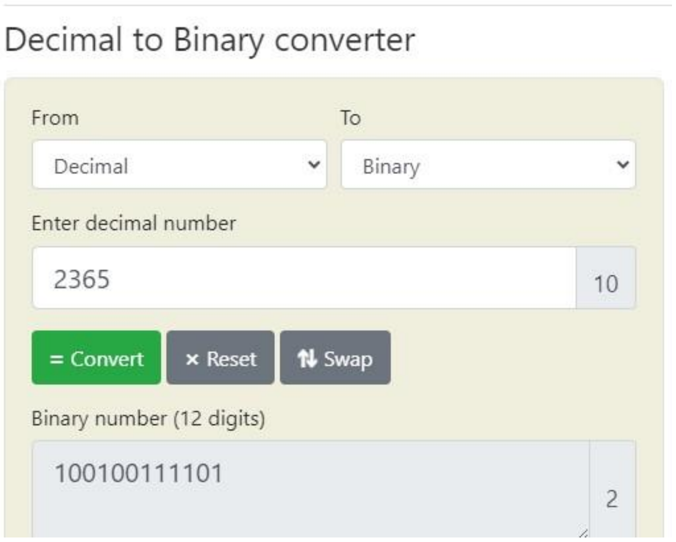
```
#include<iostream> using
namespace std;
class Stack{
        private:  int
        array[20]; int top;
        public:
        Stack(){
                top=-1;
        }
         void push(int val){
```

```
                                        ++top;
                                        array[top]=val;

          }
           void display(){


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

                                                cout<<array[i];

                                        }

                                        cout<<endl;
                                }

};


main(){

          Stack s1;

          int num, remainder,quotient;     cout<<"Enter a

decimal number: \n";

           cin>>num;


          while(num>0){

                   remainder=num%2;

                   s1.push(remainder);

                  num=num/2;

          }


          cout<<"Binary= \n";

          if(num!=0){

           cout<<num;}

           s1.display();

}
```

OUTPUT:



```
Output
/tmp/y1I8cMcdR7.o
Enter a decimal number:
2365
Binary=
100100111101
```



Decimal to Binary converter

From: Decimal
To: Binary
Enter decimal number: 2365 (10)

Binary number (12 digits): 100100111101 (2)

CODE TASK 3

Write a C++ program to check the mathematical expression is valid or not using the Stack.

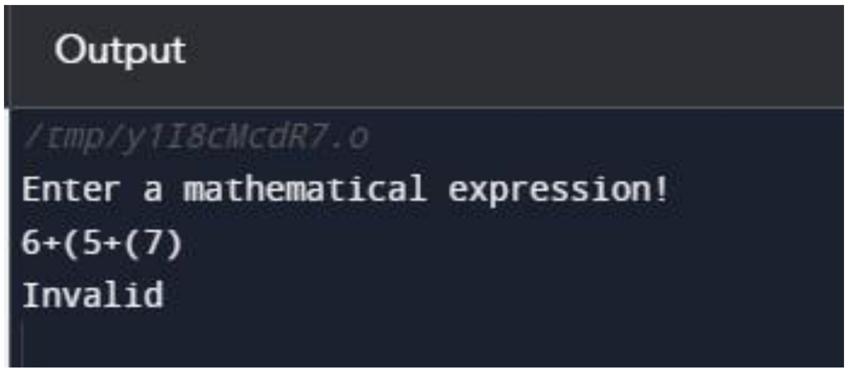
CODE:

```
#include <iostream> #include
<string.h> using namespace
std;
```

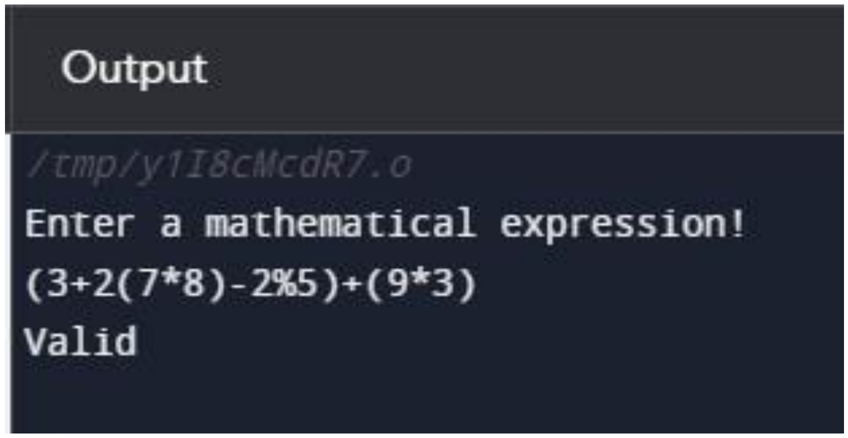
```
class Stack{   private:
        static const int size = 20;
                             char array[size];
                             int top;
               public:
                   Stack(){
                             top=-1;
                             }
        void push(char ch){
                                                         ++top;
                                                          array[top]=ch;
                                           }
                            char pop(){                       if(top>=0){
                            return array[top--];
                                           }
                                           else{
                                                cout<<"invalid\n";
                                               return '/0';
                                           }
                            }
                             bool empty(){
                                           if(top<0){
              return true;
                                           }
                                           else
                                            return false;
                            }
        void display(){
```

```
                                        if(!empty()){
                                            cout<<"Invalid\n";}
                                       else
                                        cout<<"Valid\n";
        }
};


int main() {     Stack s1;
string st;
    cout<<"Enter a mathematical expression!\n";
    getline(cin, st);

    for(int i=0; i < st.length(); i++){
        if(st[i]=='('){            s1.push('(');
        }
        else if(st[i]==')'){            s1.pop();
        }
    }
    s1.display();     return 0;
}
```

OUTPUT:



```
Output
/tmp/y1I8cMcdR7.o
Enter a mathematical expression!
6+(5+(7)
Invalid
```



```
Output
/tmp/y1I8cMcdR7.o
Enter a mathematical expression!
(3+2(7*8)-2%5)+(9*3)
Valid
```

Rabia Batool

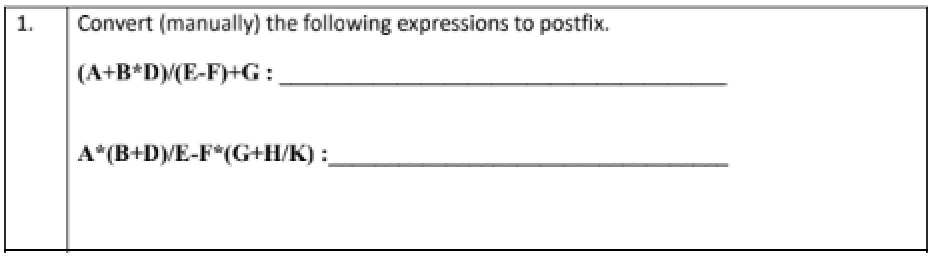
2022-BSE-064

Group#B

Data Structure & Algorithm

LAB#04

Submitted to Sir Rehan

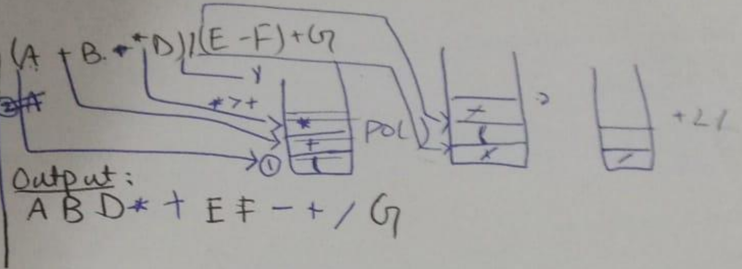


1.	Convert (manually) the following expressions to postfix. **(A+B*D)/(E-F)+G : ____________________________________** **A*(B+D)/E-F*(G+H/K) :________________________________**

Answer:

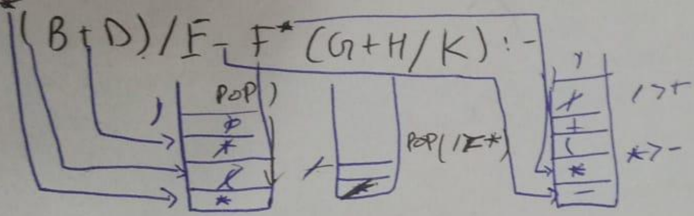
Rabia Batool
BSE-2022-064
LAB #4
Q#1

(i) ~~(A+B)~~
(i) (A + B*D)/(E − F) + G
- Operend → output
- (→ in stack
- Operators → instack
-) → pop()



Output:
A B D * + E F − + / G

② A * (B + D)/E − F * (G + H/K) :-

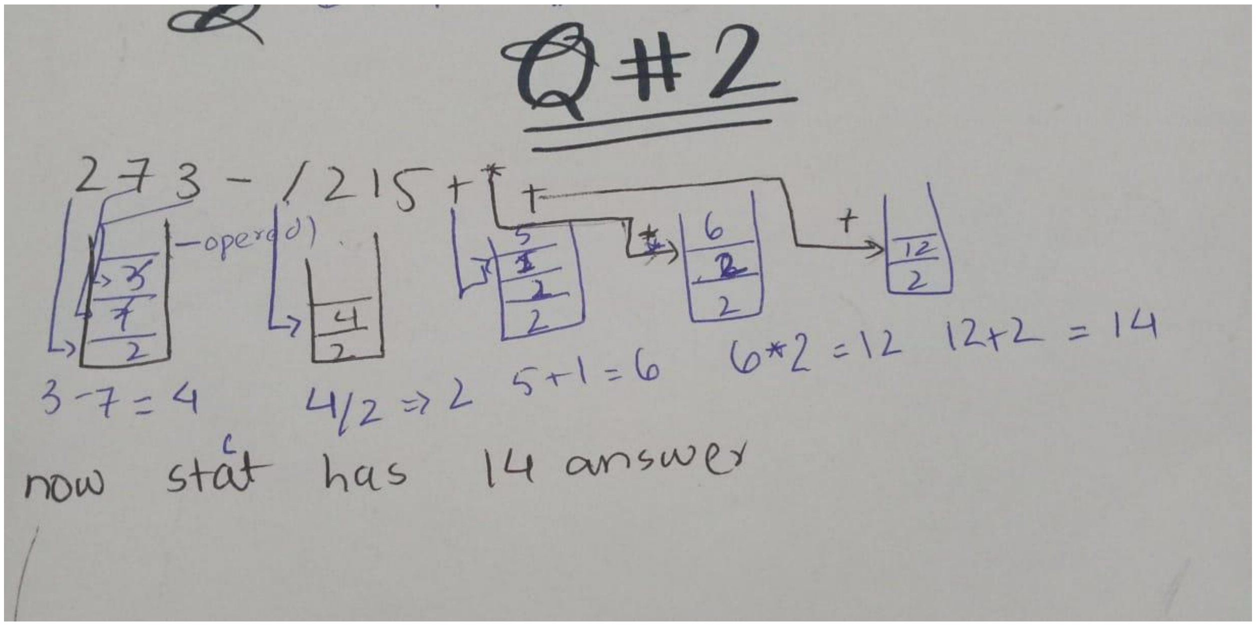


Output :-
A B D + / E* E F G H K) +

3. Convert the following expression from infix to postfix and show the contents of Stack and the output expression at each step.

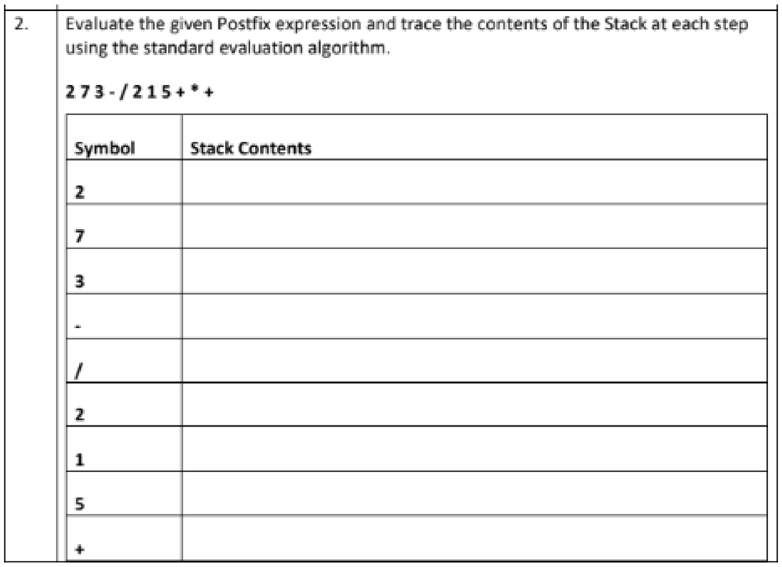
(A+B) * C – D+F*G

Symbol	Stack Contents	Output Expression
(		
A		
+		
B		
)		
*		
C		
-		
D		
+		
F		
*		
G		



3-7=4 4/2 ⇒ 2 5+1=6 6*2=12 12+2 = 14

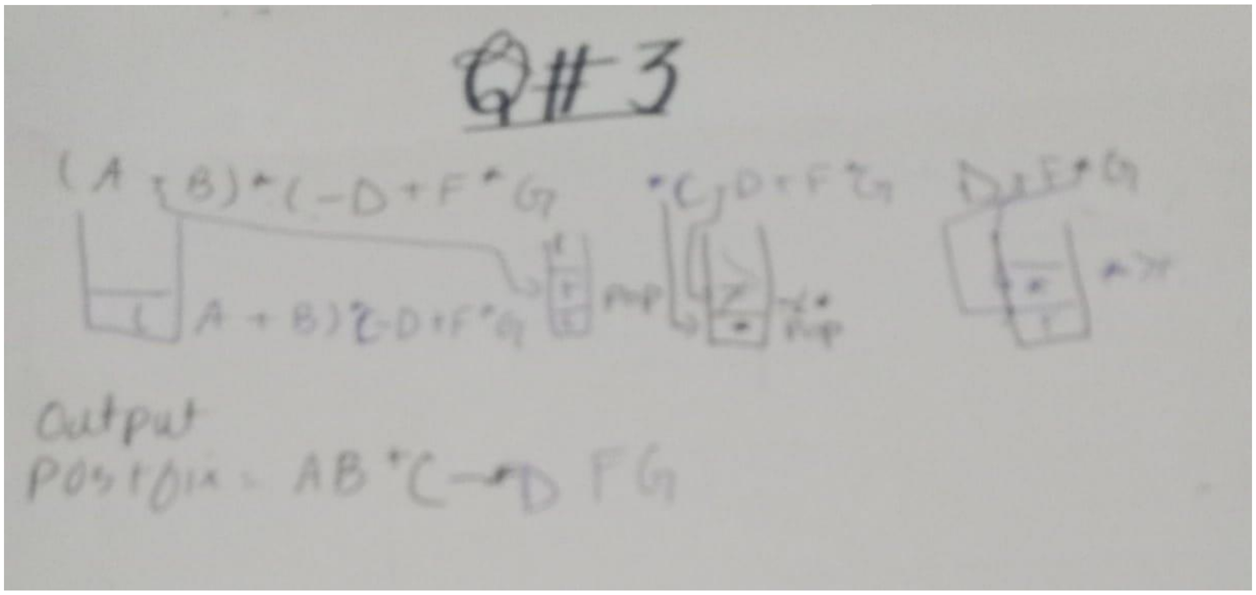
now stat has 14 answer

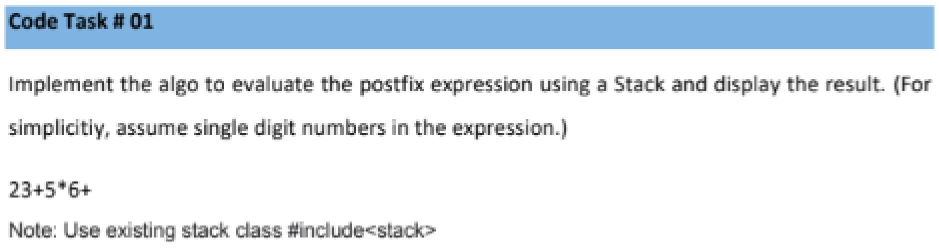


2. Evaluate the given Postfix expression and trace the contents of the Stack at each step using the standard evaluation algorithm.

2 7 3 - / 2 1 5 + * +

Symbol	Stack Contents
2	
7	
3	
-	
/	
2	
1	
5	
+	





Code Task # 01

Implement the algo to evaluate the postfix expression using a Stack and display the result. (For simplicitiy, assume single digit numbers in the expression.)

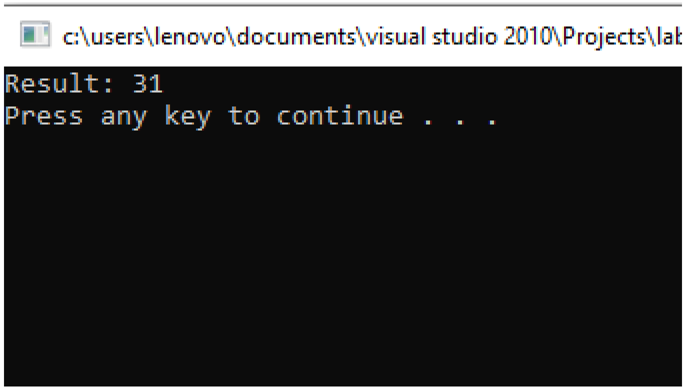
23+5*6+

Note: Use existing stack class #include<stack>

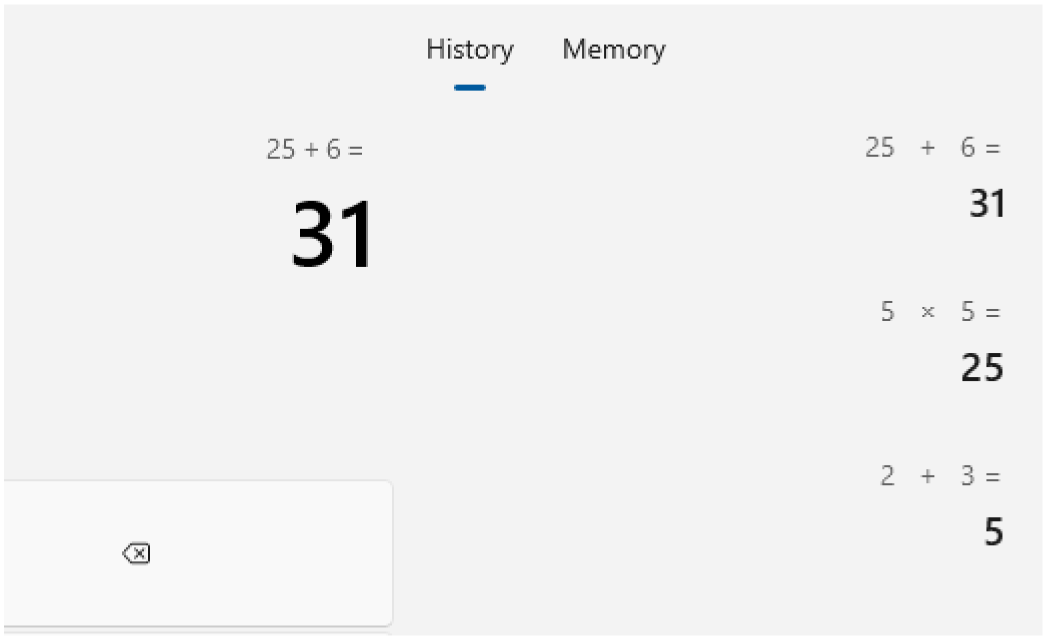
Code:

```
// lab 1.cpp : Defines the entry point for the console application.
//
```

```
#include "stdafx.h"
#include<iostream>
#include<string.h>
#include<stack>;
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{stack <int>s;

       string str;
       str="23+5*6+";
       for(int i=0; i < str.length();i++){
              if((str.at(i)=='+')||(str.at(i)=='-')||(str.at(i)=='*')||(str.at(i)=='/')){
                     int a=s.top();
                     s.pop();
                     int b=s.top();
                     s.pop();
                     if(str.at(i)=='+'){
                            int c= a+b;
                            s.push(c);
                     }
                     if(str.at(i)=='-'){
                            int c= a-b;
                            s.push(c);
                     }
                     if(str.at(i)=='*'){
                            int c= a*b;
                            s.push(c);
                     }
                     if(str.at(i)=='/'){
                            int c= a/b;
                            s.push(c);
                     }

              }
              else{
                     int x=str.at(i)-'0';

                     s.push(x);

              }



}
       cout<<"Result: "<<s.top()<<endl;

       system("pause");
       return 0;
}
```



```
Result: 31
Press any key to continue . . .
```



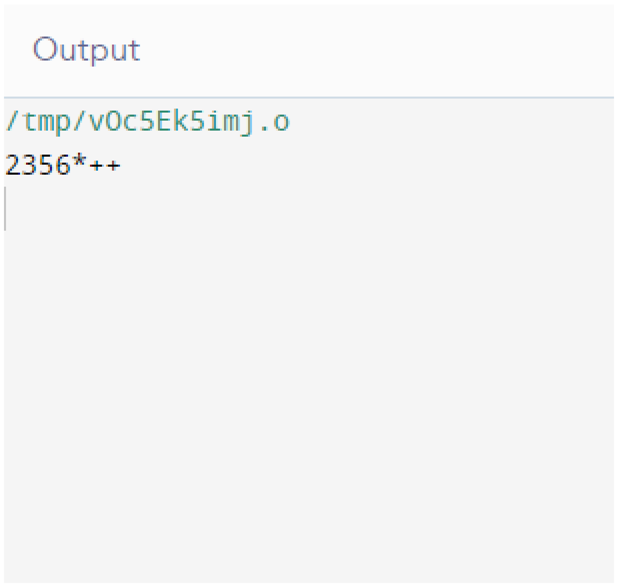
Code Task # 02

Implement the algo to covert the infix expression to postfix and display the result on screen

Note: Use existing stack class #include<stack>

```
#include <iostream>
#include <stack>
using namespace std;

int getPrecedence(char op) {
    if (op == '*' || op == '/')
        return 2;
    else if (op == '+' || op == '-')
        return 1;
    else
        return 0;
}
```

```
bool hasHigherOrEqualPrecedence(char op1, char op2) {
    return getPrecedence(op1) > getPrecedence(op2);
}

int main() {
    stack<char> s;

    string str = "(23+5*6+)";
    for (int i = 0; i < str.length(); i++) {
        if (str.at(i) == '(') {
            s.push(str.at(i));
        }
        else if (str.at(i) == ')') {
            while (!s.empty() && s.top() != '(') {
            cout<<top();
                s.pop();
            }
            s.pop();
        }
        else if (str.at(i) == '+' || str.at(i) == '-' || str.at(i) == '*' || str.at(i) ==
'/') {
            while (!s.empty() && s.top() != '(' && hasHigherOrEqualPrecedence(s.top(),
str.at(i))) {
                cout << s.top();
                s.pop();
            }
            s.push(str.at(i));
        }
        else {
            cout << str.at(i);
        }
    }

    while (!s.empty()) {
        cout << s.top();
        s.pop();
    }

    cout << endl;

    return 0;
}
```


Output

```
/tmp/vOc5Ek5imj.o
2356*++
```

Rabia Batool

2022-BSE-064

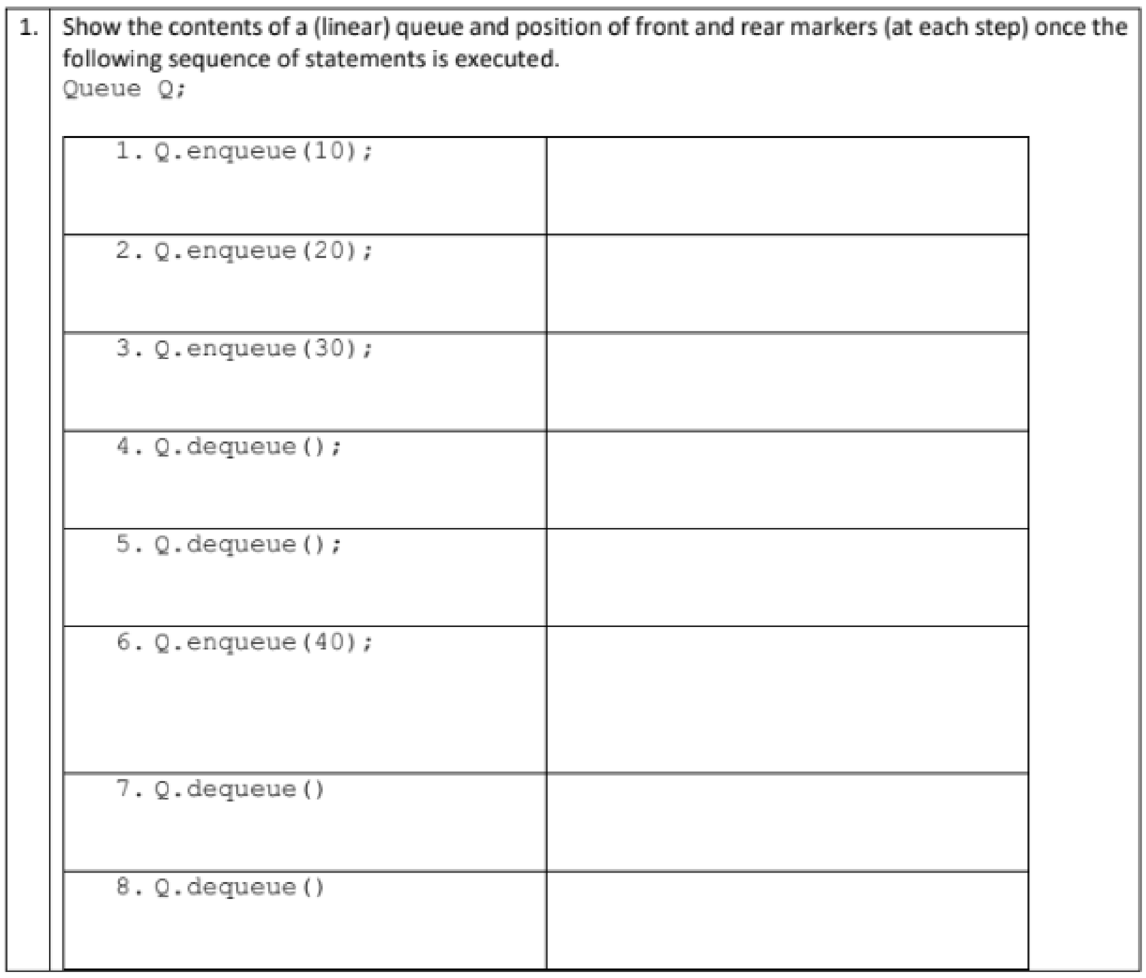
Group#B

Data Structure & Algorithm

LAB#05

Submitted to Sir Rehan

Task #1:



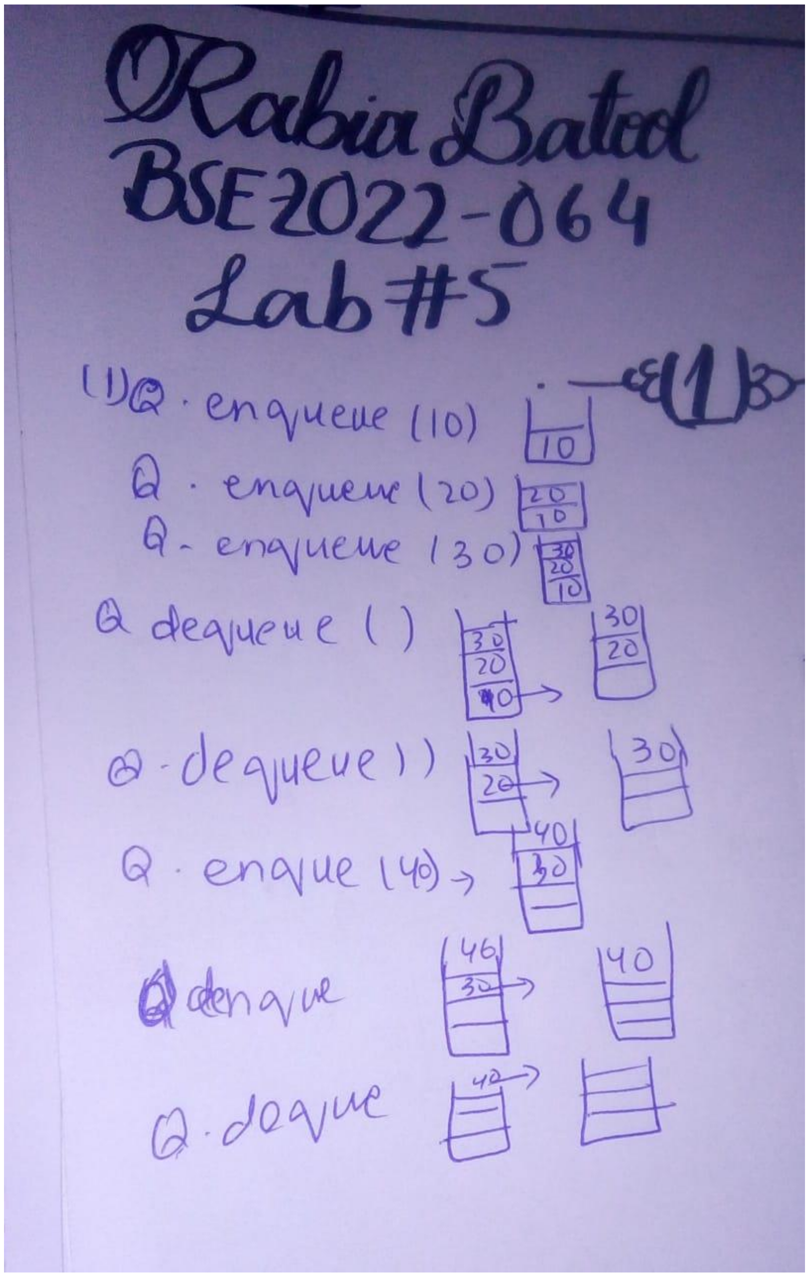
Give answers to the following.

1. Show the contents of a (linear) queue and position of front and rear markers (at each step) once the following sequence of statements is executed.

```
Queue Q;
```

`1. Q.enqueue(10);`	
`2. Q.enqueue(20);`	
`3. Q.enqueue(30);`	
`4. Q.dequeue();`	
`5. Q.dequeue();`	
`6. Q.enqueue(40);`	
`7. Q.dequeue()`	
`8. Q.dequeue()`	

Answer:



Rabia Batool

BSE2022-064

Lab#5

(1)

(1)Q. enqueue (10) 10

Q. enqueue (20) 20 10

Q. enqueue (30) 30 20 10

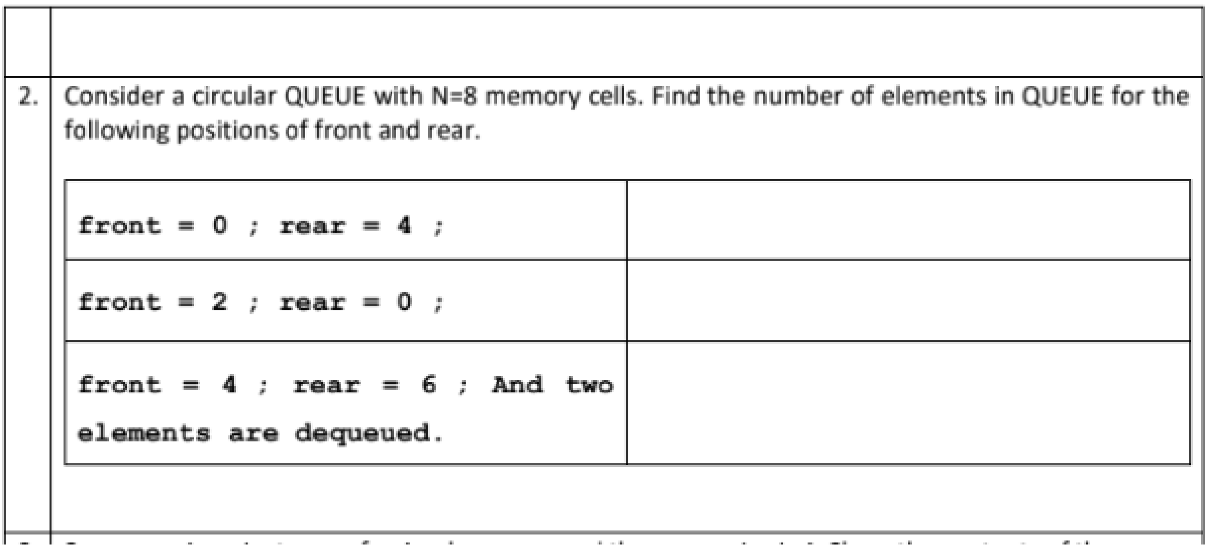
Q dequeue ( ) 30 20 10 → 30 20

Q. dequeue ( ) 30 20 → 30

Q. enque (40) → 40 30

Q deque 40 30 → 40

Q. deque 40 →



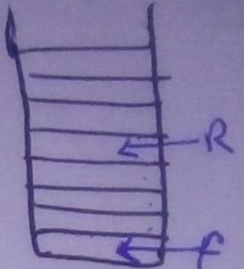
2.	Consider a circular QUEUE with N=8 memory cells. Find the number of elements in QUEUE for the following positions of front and rear.

`front = 0 ; rear = 4 ;`	
`front = 2 ; rear = 0 ;`	
`front = 4 ; rear = 6 ; And two elements are dequeued.`	

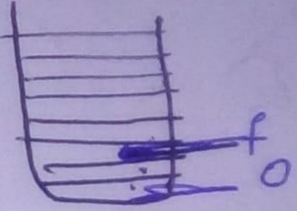
Answer:

(2)

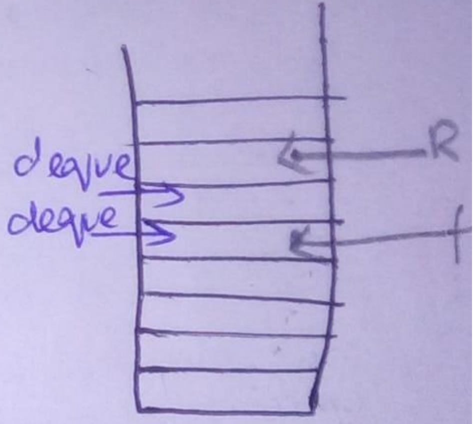
(i) front = 0 ; rear = 4
no. of elenent = 4



(ii) front = 2; rear = 0
no. of element = 1

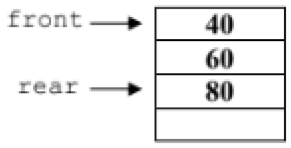


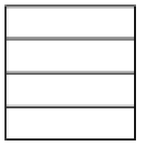
(iii) front 4, reer 6
no. of elenents
after deque = 0

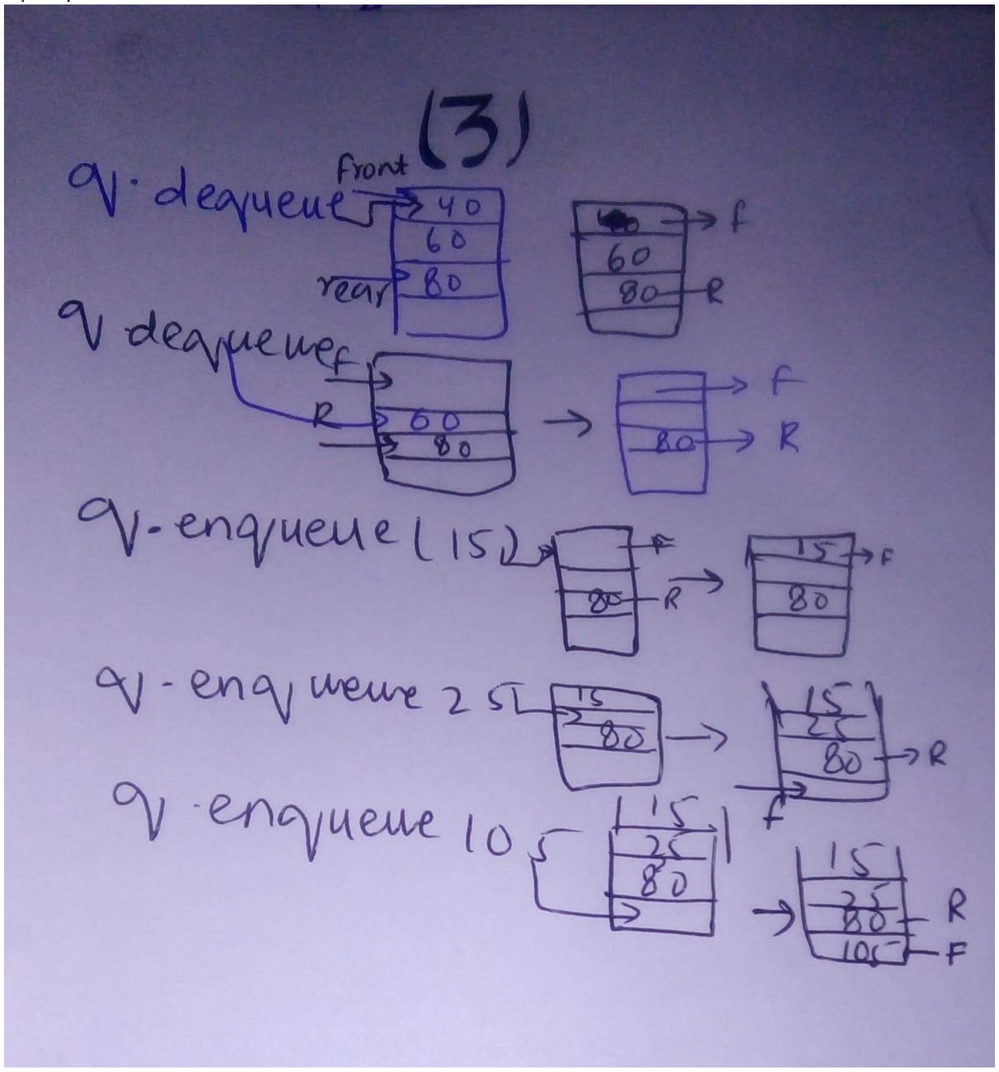


3. Suppose q is an instance of a circular queue and the queue size is 4. Show the contents of the queue and positions of the front and rear markers once the following sequence of statements is executed. The initial contents of the queue are listed in the following.

```
q.dequeue();
q.dequeue();
q.enqueue(15);
q.enqueue(25);
q.enqueue(105);
```





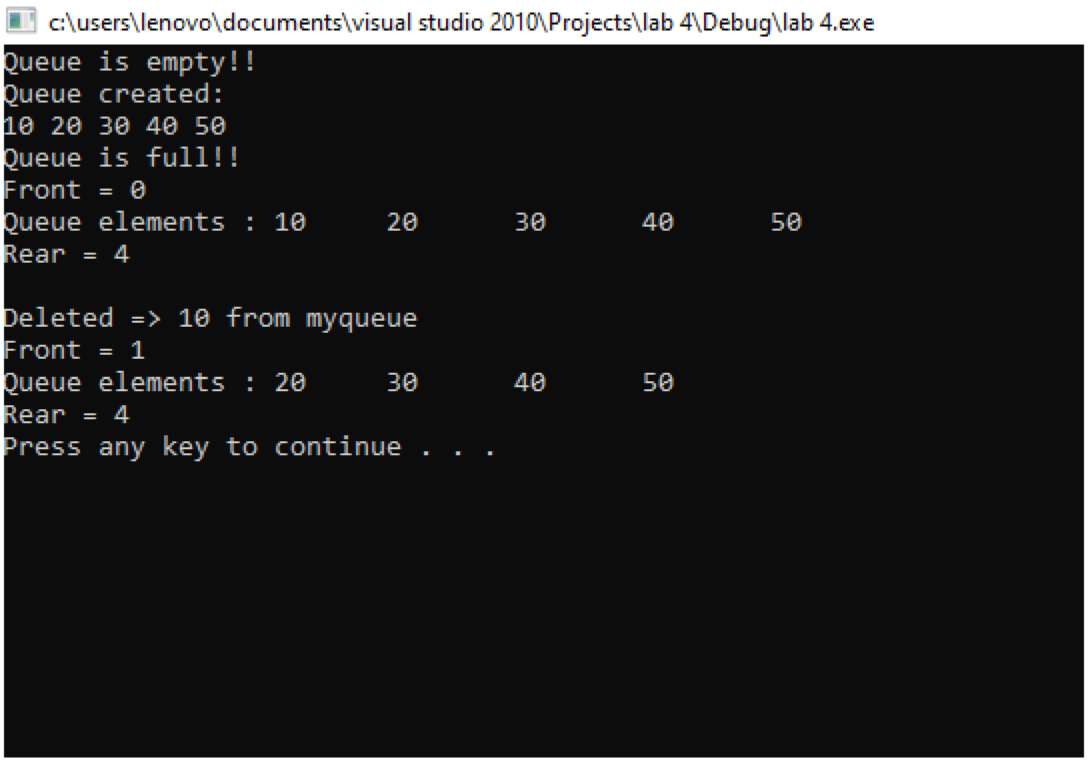


Code Task # 01

Create a class Queue that implements the functionality of a queue providing all the required operations (Addqueue(), Removequeue(), is_Empty(), is_Full(),display() , getFront(), getRare()).

```
// lab 4.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <iostream>
using namespace std;
#define MAX_SIZE 5
using namespace std;

class Queue {
private:
int myqueue[MAX_SIZE], front, back;

public:
Queue(){
front = -1;
back = -1;
    }

bool isFull(){
if(front == 0 && back == MAX_SIZE - 1){
return true;
        }
return false;
    }

bool isEmpty(){
if(front == -1) return true;
else
       return false;
    }

void addqueue(int value){
if(isFull()){
cout << endl<< "Queue is full!!";
        }
else {
if(front == -1)
       front = 0;
back++;
myqueue[back] = value;
cout << value << " ";
        }
    }
int removequeue(){
int value;
if(isEmpty()){
cout << "Queue is empty!!" << endl; return(-1); }
else { value = myqueue[front];
 if(front >= back){      //only one element in queue
front = -1;
back = -1;
            }
else {
front++;
            }
cout << endl << "Deleted => " << value << " from myqueue";
return(value);
        }
    }
int getfront(){
       if (front!=-1)
              return myqueue[front];}
int getback(){
       if(back!=-1)
              return myqueue[back]};

    /* Function to display elements of Queue */
void displayQueue()
    {
int i;
if(isEmpty()) {
cout << endl << "Queue is Empty!!" << endl;
        }
else {
cout << endl << "Front = " << front;
cout << endl << "Queue elements : ";
for(i=front; i<=back; i++)
cout << myqueue[i] << "\t";
cout << endl << "back = " << back << endl;
        }
    }
};

int _tmain(int argc, _TCHAR* argv[])
{Queue myq;
myq.removequeue();      //deQueue
myq.isEmpty();
cout<<"Queue created:"<<endl;
myq.addqueue(10);
myq.addqueue(20);
int c=myq.getfront();
cout<<c;
myq.addqueue(30);
myq.addqueue(40);
myq.addqueue(50); //enqueue 60 => queue is full
myq.addqueue(60);
myq.isFull();
myq.displayQueue();

    //deQueue =>removes 10
myq.removequeue();

    //queue after dequeue
```

```
myq.displayQueue();
system("pause");
       return 0;
}
```



c:\users\lenovo\documents\visual studio 2010\Projects\lab 4\Debug\lab 4.exe

```
Queue is empty!!
Queue created:
10 20 30 40 50
Queue is full!!
Front = 0
Queue elements : 10     20      30      40      50
Rear = 4

Deleted => 10 from myqueue
Front = 1
Queue elements : 20     30      40      50
Rear = 4
Press any key to continue . . .
```

Code Task # 02

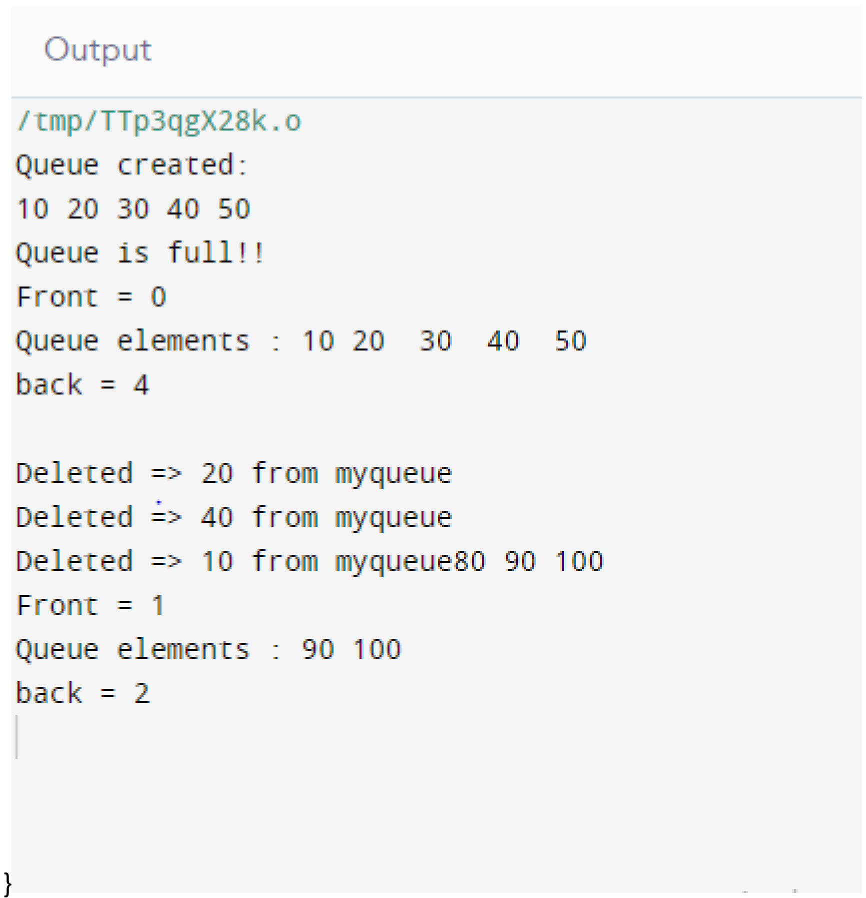
Create a class Circular Queue that implements the functionality of a queue providing all the required operations (Addqueue(), Removequeue(), Empty(), Full() and getFront()).

```
// Online C++ compiler to run C++ program online

// lab 4.cpp : Defines the entry point for the console application.

//



#include <iostream>

using namespace std;

#define MAX_SIZE 5
```

```
using namespace std;

class Queue {
private:
int myqueue[MAX_SIZE], front, back;

public:
Queue(){
front = -1;
back = -1;
   }

bool isFull(){
if(front == 0 && back == MAX_SIZE - 1){
return front == (back + 1) % MAX_SIZE;
       }
return false;
   }

bool isEmpty(){
if(front == -1) return true;
else
            return false;
   }

void addqueue(int value){
if(isFull()){
cout << endl<< "Queue is full!!";
       }
```

```
else {
if(front == -1)
            front = 0;
back = (back + 1) % MAX_SIZE;
myqueue[back] = value;
cout << value << " ";
       }
   }
int removequeue(){
int value;
if(isEmpty()){
cout << "Queue is empty!!" << endl; return(-1); }
else { front = (front + 1) % MAX_SIZE;
            value = myqueue[front];
 if(front >= back){     //only one element in queue
front = -1;
back = -1;
          }
else {
front++;
          }
cout << endl << "Deleted => " << value << " from myqueue";
return(value);
       }
   }

   /* Function to display elements of Queue */
void displayQueue()
   {
```

```
int i;
if(isEmpty()) {
cout << endl << "Queue is Empty!!" << endl;
        }
else {
cout << endl << "Front = " << front;
cout << endl << "Queue elements : ";
for(i=front; i<=back; i++)
cout << myqueue[i] << "\t";
cout << endl << "back = " << back << endl;
        }
    }
};

int main()
{Queue myq;
cout<<"Queue created:"<<endl;
myq.addqueue(10);
myq.addqueue(20);
myq.addqueue(30);
myq.addqueue(40);
myq.addqueue(50); //enqueue 60 => queue is full
myq.addqueue(60);
myq.displayQueue();

    //deQueue =>removes 10
myq.removequeue();
myq.removequeue();
myq.removequeue();
myq.addqueue(80);
myq.addqueue(90);
myq.addqueue(100);
   //queue after dequeue
myq.displayQueue();
system("pause");
            return 0;
```



Output

```
/tmp/TTp3qgX28k.o
Queue created:
10 20 30 40 50
Queue is full!!
Front = 0
Queue elements : 10 20  30  40  50
back = 4

Deleted => 20 from myqueue
Deleted => 40 from myqueue
Deleted => 10 from myqueue80 90 100
Front = 1
Queue elements : 90 100
back = 2
```

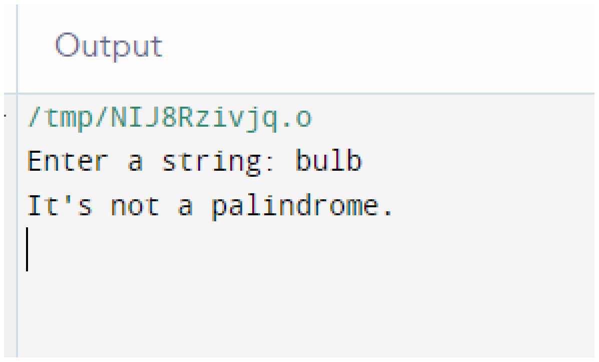
```
}
```

Code Task # 03

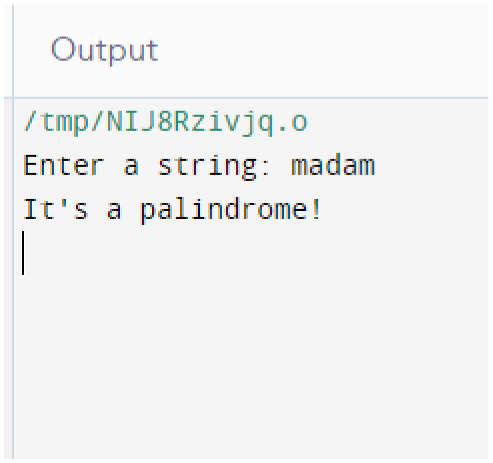
Use **Stack AND Queue** combination to find weather a word is a PALINDROME or not as discussed in Lab hours

```
#include <iostream>
```

```
#include <iostream>

#include <stack>

#include <queue>

#include <string>

#include <cctype> // For tolower


using namespace std;


bool isPalindrome(string s) {

    stack<char> charStack;

    queue<char> charQueue;


    // Convert the input string to lowercase and remove spaces

    for (char c : s) {

        if (isalnum(c)) { // Check if the character is alphanumeric

            charStack.push(tolower(c));

            charQueue.push(tolower(c));

        }

    }


    while (!charStack.empty() && !charQueue.empty()) {

        if (charStack.top() != charQueue.front()) {

            return false;

        }


        charStack.pop();

        charQueue.pop();

    }
    return true;
}

int main() {
    string input;
    cout << "Enter a string: ";
    getline(cin, input);

    if (isPalindrome(input)) {
        cout << "It's a palindrome!" << endl;
    } else {
        cout << "It's not a palindrome." << endl;
    }

    return 0;
}
```


Output

```
/tmp/NIJ8Rzivjq.o
Enter a string: bulb
It's not a palindrome.
```



```
/tmp/NIJ8Rzivjq.o
Enter a string: madam
It's a palindrome!
```

Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

Data Structures and Algorithms

Lab 6

Submitted to:

Sir Rehan Ahmed

Submitted by:

Rabiabatool

2022BSE064

Task 4

1\.

The following list of names is assigned (in order) to a linear array INFO. Assign value to LINK and START, so that INFO, LINK and START form an alphabetical list.

START

Q

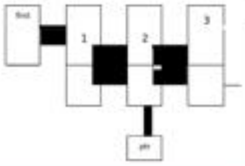
W E R T Y U I O P

INFO	LINK
Mary	
Helen	
Barbara	
Paula	
Diana	
Audrey	
Karen	
Nancy	
Ruth	
Eileen	

Rabya Batool
064
(1) [illegible]

2\.

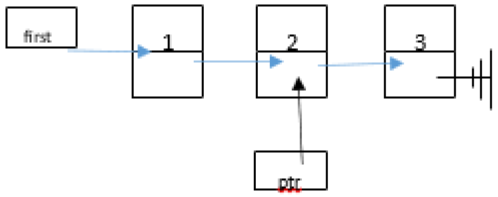
Given the following linked list, state what does each of the following statements refer to.



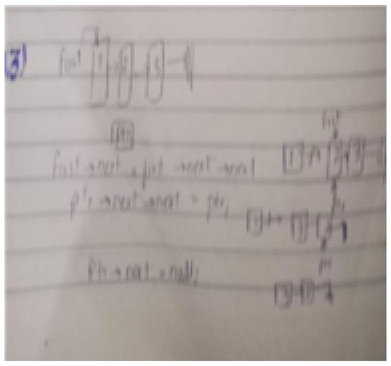
(2) [illegible]

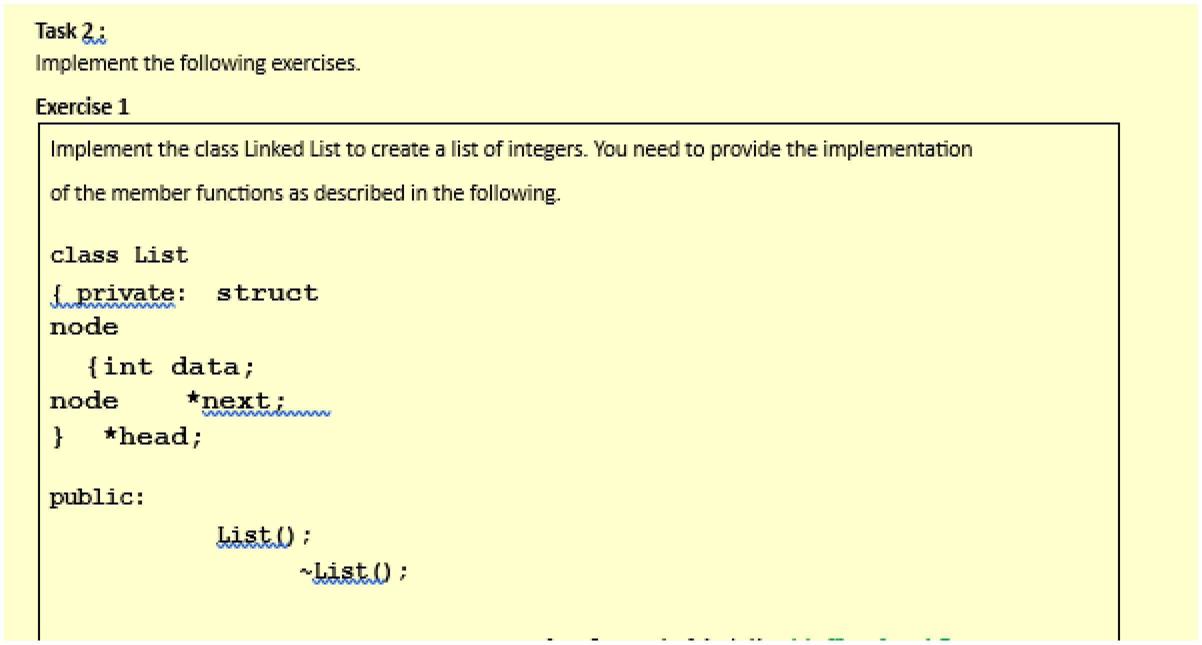
`first->data;`	1
`first->next->next->data;`	2
`ptr->next->data;`	3
`ptr->next->next;`	null
`first->next->next->next;`	3 node address

3. he following list after the given instructions are executed:



```
first -> next = first -> next -> next;
ptr -> next -> next = ptr;
ptr->next = NULL;
```





Task 2 :

Implement the following exercises.

Exercise 1

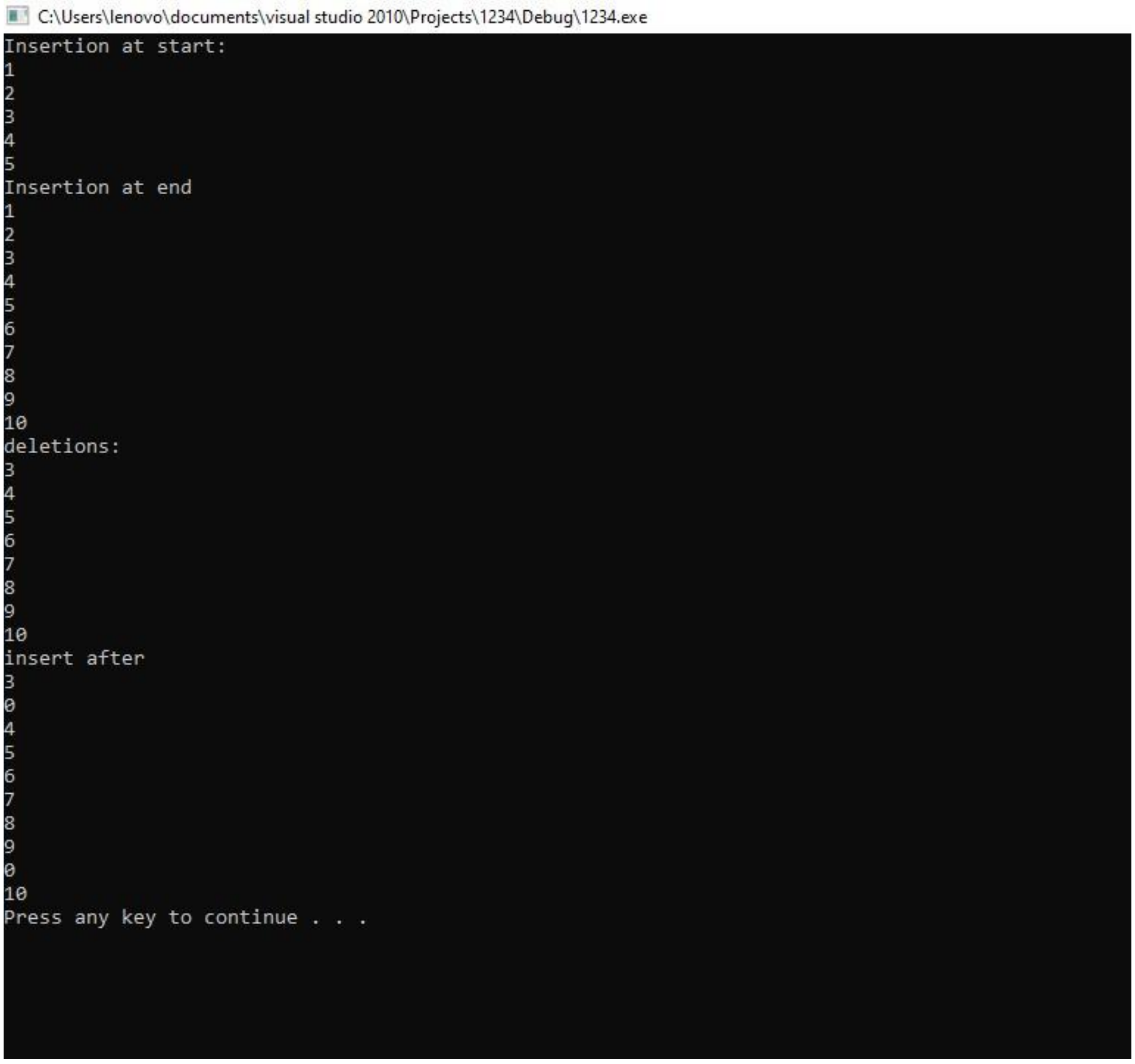
Implement the class Linked List to create a list of integers. You need to provide the implementation of the member functions as described in the following.

```
class List
{ private:  struct
node
   {int data;
node     *next;
}   *head;

public:
             List();
                    ~List();
```

```
//

#include "stdafx.h"
#include<iostream> using
namespace std;

 class List { private: struct node {int data; node *next; } *head; public: List()

{head=NULL;

}  bool
emptyList()
{if(head==NULL)
return true;

else

            return false;}

void insertafter(int oldV, int newV) {
node* temp = new node;     temp->data
= newV;     temp->next = NULL;


   if (head == NULL) {

      head = temp;

          } else if(oldV<head->data)
{node*ptr=new node;
```

```
ptr->data=newV; ptr-
>next=NULL; ptr-
>next=head; head=ptr;

            }

            else

             {node *ptr;

    ptr = head; while (ptr-
>data != oldV)

{ptr = ptr->next;

}

temp -> next = ptr -> next; ptr ->
next = temp; }

}

 void deleteNode(int value)

{int flag=0; node
*s1,*s2,*temp;

            if(head==NULL)

{cout<<"linklist is empty"<<endl;} else
if(head->data=value)

{node *temp=head;
head=head->next; temp-
>next=NULL; delete
temp;

            }

else { s1=head; s2=s1-
>next; while(s2-
>next!=NULL) {if(s2-
>data==value)
{temp=s2; s2=temp-
>next; s1->next=s2;
temp->next=NULL;
delete temp; flag++;}
else {s1=s1->next;
s2=s2->next;} }
if(flag==0) {temp=s1-
>next; s1->next=NULL;
delete temp;}}





}

void insert_begin(int value)
{node*ptr=new node; ptr-
>data=value; ptr->next=NULL; ptr-
>next=head; head=ptr;  } void
insert_end(int value) {     node*
temp = new node;     temp->data =
value;     temp->next = NULL;      if
(head == NULL) {         head = temp;
} else {         node* s = head;
while (s->next != NULL) {             s = s-
>next;         }         s->next = temp;

    }

}


void traverse() {node
*ptr=head;

             while (ptr!=NULL)
{cout<<ptr->data<<endl;

                           ptr=ptr->next;

             }

}


};


int _tmain(int argc, _TCHAR* argv[])

{

 List l; cout<<"Insertion at
start:"<<endl;
l.insert_begin(5);

l.insert_begin(4);
```

```
l.insert_begin(3);
l.insert_begin(2);
l.insert_begin(1);
l.traverse(); cout<<"Insertion at
end"<<endl;
l.insert_end(6);
l.insert_end(7);
l.insert_end(8);
l.insert_end(9);
l.insert_end(10);
l.traverse(); cout<<"deletions:"<<endl;
l.deleteNode(1);
l.deleteNode(2);
l.traverse(); cout<<"insert
after"<<endl;
l.insertafter(3,0);
l.insertafter(9,0);
l.traverse();
        system("pause");
        return 0;
}
```


C:\Users\lenovo\documents\visual studio 2010\Projects\1234\Debug\1234.exe

```
Insertion at start:
1
2
3
4
5
Insertion at end
1
2
3
4
5
6
7
8
9
10
deletions:
3
4
5
6
7
8
9
10
insert after
3
0
4
5
6
7
8
9
0
10
Press any key to continue . . .
```

Fatima Jinnah Women University

Rabia Batool

2022-BSE-064

Group#B

Data Structure & Algorithm

LAB#07

Submitted to Sir Rehan

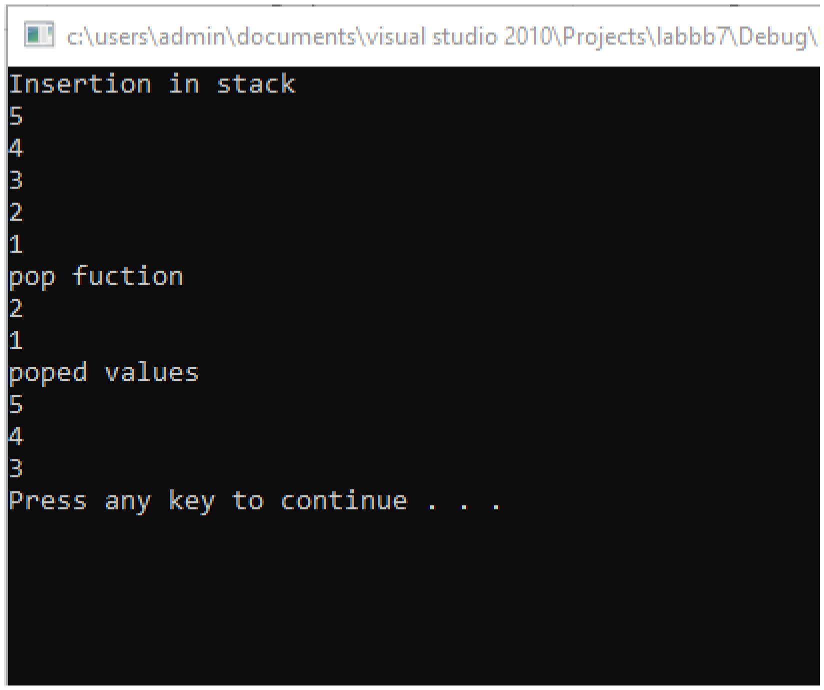
Exercise 1

A stack can be implemented using a linked list. The first node can serve as the 'top' of Stack and 'push' and 'pop' operations can be implemented by adding and removing nodes at the head of the linked list. Implement the Stack class using a linked list and provide all the standard member functions.

```cpp
// labbb7.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;


class stack
{
```

```
private:
struct node
{int data;
node *next;
} *top;
public:
stack()
{top=NULL;
}
bool emptyList()
{if(top==NULL)
return true;
else
return false;}

int pop()
{
node **temp;
if(top==NULL)
{cout<<"linklist is empty"<<endl;
return 0;
}
else
       {node *temp=top;
top=temp->next;
temp->next=NULL;
int x;
x=temp->data;
delete temp;
return x;

}}

void push(int value)
{node*ptr=new node;
ptr->data=value;
ptr->next=NULL;
ptr->next=top;
top=ptr;
}

void traverse()
{node *ptr=top;
while (ptr!=NULL)
{cout<<ptr->data<<endl;
ptr=ptr->next;
}
}
};
int _tmain(int argc, _TCHAR* argv[])
{
       stack l;
cout<<"Insertion in stack"<<endl;
l.push(1);
l.push(2);
l.push(3);
l.push(4);
l.push(5);
```

```
l.traverse();
cout<<"pop fuction"<<endl;
int a,b,c;
a=l.pop();
b=l.pop();
c=l.pop();
l.traverse();
cout<<"poped values"<<endl;
cout<<a<<endl;
cout<<b<<endl;;
cout<<c<<endl;
       system("pause");
       return 0;
}
```



c:\users\admin\documents\visual studio 2010\Projects\labbb7\Debug\

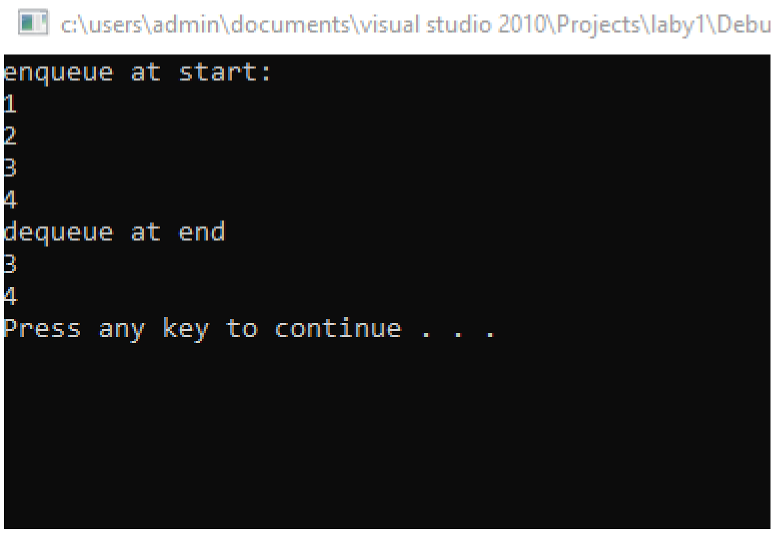
```
Insertion in stack
5
4
3
2
1
pop fuction
2
1
poped values
5
4
3
Press any key to continue . . .
```

Exercise 2

Implement simple (Linear) Queue through linked list.

```
// laby1.cpp : Defines the entry point for the console application.
//

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

```
using namespace std;


class queue
{
private:
struct node
{int data;
node *next;
} *front,*rear;
public:
queue()
{front=NULL;
rear=NULL;}
bool emptyList()
{if(front==NULL)
return true;
else
return false;}

 void dequeue()
    {
        if (front == NULL)
        {
            cout << "Queue is empty" << endl;
        }
        else
        {
            node* temp = front;
            front = front->next;
            delete temp;

            if (front == NULL)
            {
                rear = NULL;
            }
        }
    }

    void enqueue(int value)
    {
        node* temp = new node;
        temp->data = value;
        temp->next = NULL;

        if (front == NULL)
        {
            front = temp;
            rear = temp;
        }
        else
        {
            rear->next = temp;
            rear = temp;
        }
    }

void traverse()
{node *ptr=front;
while (ptr!=NULL)
{cout<<ptr->data<<endl;
ptr=ptr->next;
}
}
};
int _tmain(int argc, _TCHAR* argv[])
{queue l;
cout<<"enqueue at start:"<<endl;
l.enqueue(1);
l.enqueue(2);
l.enqueue(3);
l.enqueue(4);
l.traverse();
cout<<"dequeue at end"<<endl;
l.dequeue();
l.dequeue();
l.traverse();

       system("pause");
       return 0;
}
```



```
c:\users\admin\documents\visual studio 2010\Projects\laby1\Debu
```

```
enqueue at start:
1
2
3
4
dequeue at end
3
4
Press any key to continue . . .
```

Exercise 3

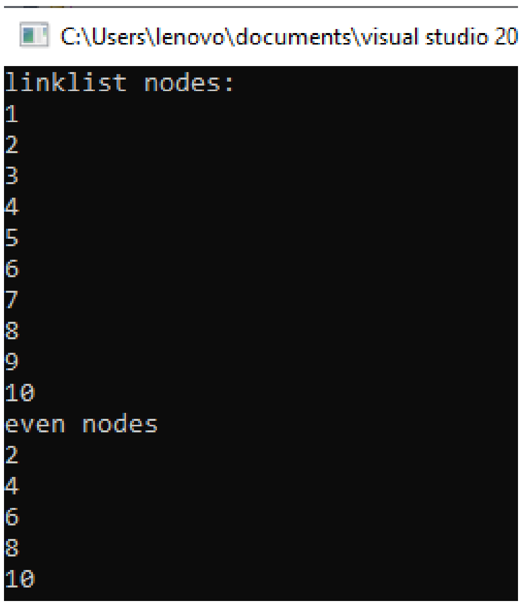
Write the following C++ functions to realize the indicated functionality on a singly linked list of integers.

a. A function that prints only the even-numbered nodes of the list.
b. A function which takes two data values as arguments and swaps the node values containing them.
c. A function that deletes the first half of the linked list nodes. You are required to first determine the total number of linked list nodes and then delete the first part of the linked list.

```
// lab 7.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;


class List
{
private:
struct node
{int data;
node *next;
} *head;
public:
List()
{head=NULL;
}

bool emptyList()
{if(head==NULL)
return true;
else
       return false;}

void even_node()
{node*temp=head->next;
while(temp!=NULL)
{cout<<temp->data;
cout<<endl;
       temp=temp->next->next;
}}
void insert_begin(int value)
```

```
{node*ptr=new node;
ptr->data=value;
ptr->next=NULL;
ptr->next=head;
head=ptr;

}
void insert_end(int value)
{
    node* temp = new node;
    temp->data = value;
    temp->next = NULL;

    if (head == NULL) {
        head = temp;
    } else {
        node* s = head;
        while (s->next != NULL) {
            s = s->next;
        }
        s->next = temp;
    }
}

void traverse()
{node *ptr=head;
       while (ptr!=NULL)
       {cout<<ptr->data<<endl;
              ptr=ptr->next;
       }
}

};



int _tmain(int argc, _TCHAR* argv[])
{
       List l;
cout<<"linklist nodes:"<<endl;
l.insert_begin(5);
l.insert_begin(4);
l.insert_begin(3);
l.insert_begin(2);
l.insert_begin(1);
l.insert_end(6);
l.insert_end(7);
l.insert_end(8);
l.insert_end(9);
l.insert_end(10);
l.traverse();
cout<<"even nodes"<<endl;
l.even_node();
l.even_node();
```

```
        system("pause");

        return 0;
}
```

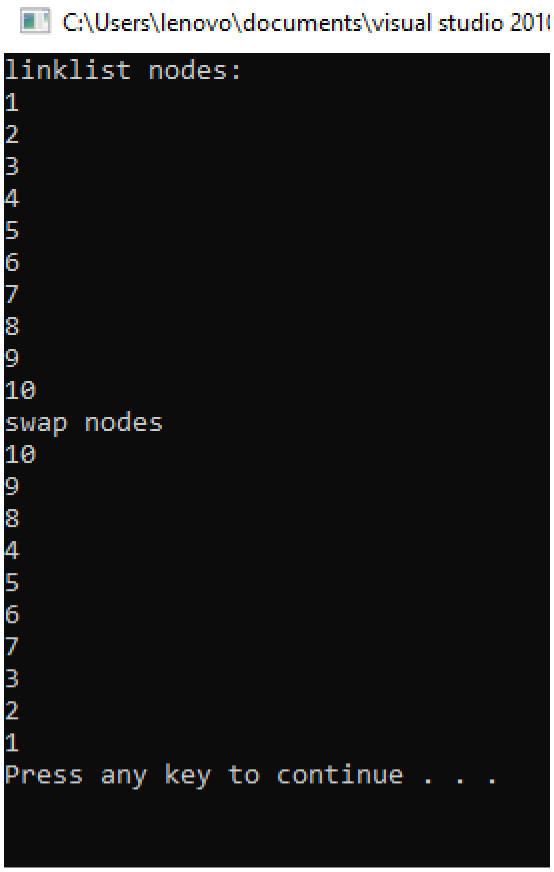


C:\Users\lenovo\documents\visual studio 20

```
linklist nodes:
1
2
3
4
5
6
7
8
9
10
even nodes
2
4
6
8
10
```

(B)

```
// lab 7.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;


class List
{
private:
struct node
{int data;
node *next;
} *head;
public:
List()
{head=NULL;
}

bool emptyList()
{if(head==NULL)
return true;
else
        return false;}
void swap(int a, int b)
{
    if (a == b) {
        // No need to swap if a and b are the same.
        return; }

    node* t1 = head;
    node* t2 = head;
    node* temp1 = NULL;
    node* temp2 =NULL;

    while (t1 != NULL && t1->data != a) {
        temp1 = t1;
        t1 = t1->next; }
 while (t2 != NULL && t2->data != b) {
        temp2 = t2;
        t2 = t2->next;}
if (t1 == NULL || t2 == NULL) {
        cout<<"null";}
if (temp1 != nullptr) {
        temp1->next = t2; }
else {
        head = t2;}
if (temp2 != nullptr) {
        temp2->next = t1;}
else {
        head = t1;}

    node* temp = t1->next;
    t1->next = t2->next;
    t2->next = temp;
}


void insert_begin(int value)
{node*ptr=new node;
ptr->data=value;
ptr->next=NULL;
ptr->next=head;
head=ptr;

}
void insert_end(int value)
{
    node* temp = new node;
    temp->data = value;
    temp->next = NULL;

    if (head == NULL) {
        head = temp;
    } else {
        node* s = head;
        while (s->next != NULL) {
            s = s->next;
        }
        s->next = temp;
    }
}

void traverse()
{node *ptr=head;
       while (ptr!=NULL)
       {cout<<ptr->data<<endl;
              ptr=ptr->next;
       }
}

};



int _tmain(int argc, _TCHAR* argv[])
{
       List l;
cout<<"linklist nodes:"<<endl;
l.insert_begin(5);
l.insert_begin(4);
l.insert_begin(3);
l.insert_begin(2);
l.insert_begin(1);
l.insert_end(6);
l.insert_end(7);
l.insert_end(8);
l.insert_end(9);
l.insert_end(10);
l.traverse();
cout<<"swap nodes"<<endl;
l.swap(1,10);
l.swap(3,8);
l.swap(2,9);
l.traverse();


       system("pause");

       return 0;
}
```



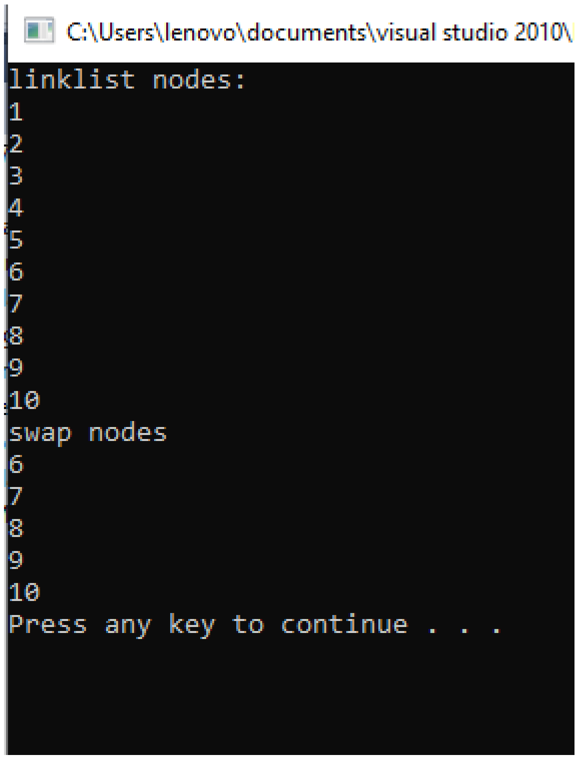
```
C:\Users\lenovo\documents\visual studio 201
linklist nodes:
1
2
3
4
5
6
7
8
9
10
swap nodes
10
9
8
4
5
6
7
3
2
1
Press any key to continue . . .
```

(C)

```
// lab 7.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;


class List
{
private:
struct node
{int data;
node *next;
} *head;
public:
List()
{head=NULL;
}

bool emptyList()
{if(head==NULL)
return true;
else
```

```
       return false;}
void delete_half()
{int count=0;
node*temp=head;
while (temp!=NULL)
{count ++;
temp=temp->next;}
int half=0;
half=count/2;
for(int i=0;i<half;i++)
{node*ptr=head;
head=head->next;
ptr->next =NULL;
delete ptr;}

}



void insert_begin(int value)
{node*ptr=new node;
ptr->data=value;
ptr->next=NULL;
ptr->next=head;
head=ptr;

}
void insert_end(int value)
{
    node* temp = new node;
    temp->data = value;
    temp->next = NULL;

    if (head == NULL) {
        head = temp;
    } else {
        node* s = head;
        while (s->next != NULL) {
            s = s->next;
        }
        s->next = temp;
    }
}

void traverse()
{node *ptr=head;
       while (ptr!=NULL)
       {cout<<ptr->data<<endl;
              ptr=ptr->next;
       }
```

```
}

};



int _tmain(int argc, _TCHAR* argv[])
{
       List l;
cout<<"linklist nodes:"<<endl;
l.insert_begin(5);
l.insert_begin(4);
l.insert_begin(3);
l.insert_begin(2);
l.insert_begin(1);
l.insert_end(6);
l.insert_end(7);
l.insert_end(8);
l.insert_end(9);
l.insert_end(10);
l.traverse();
cout<<"swap nodes"<<endl;
l.delete_half();
l.traverse();


       system("pause");

       return 0;
}
```



C:\Users\lenovo\documents\visual studio 2010\

```
linklist nodes:
1
2
3
4
5
6
7
8
9
10
swap nodes
6
7
8
9
10
Press any key to continue . . .
```

Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

DATA STRUCTURE & ALGORITHM

LAB # 08

Submitted to:

Sir rehan

Submitted by:

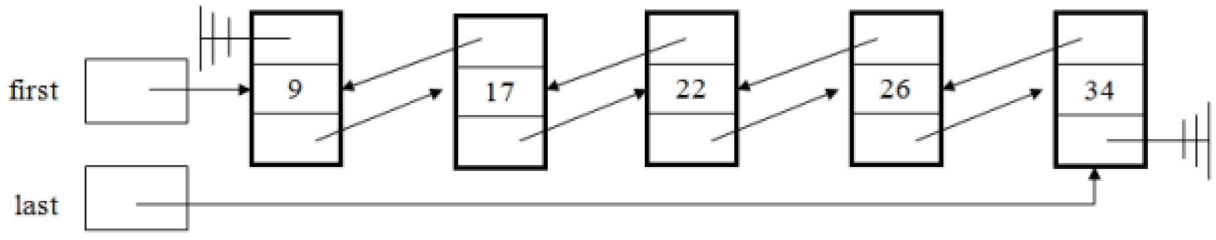
Rabia Batool

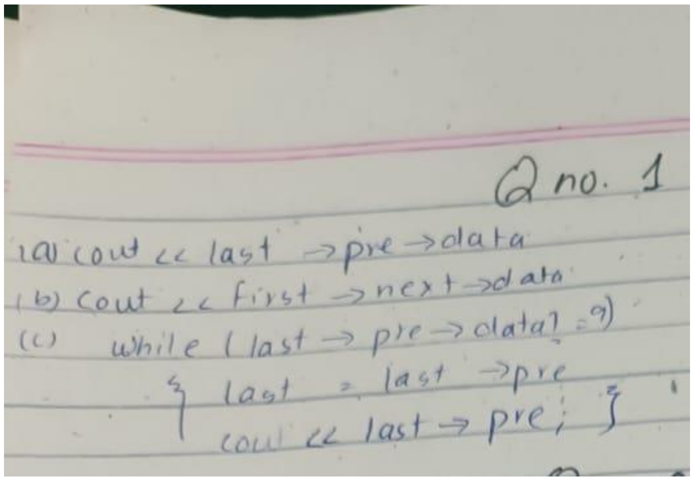
2022-BSE-064

Task 1 :

Give answers to the following.

Consider the following doubly linked list.





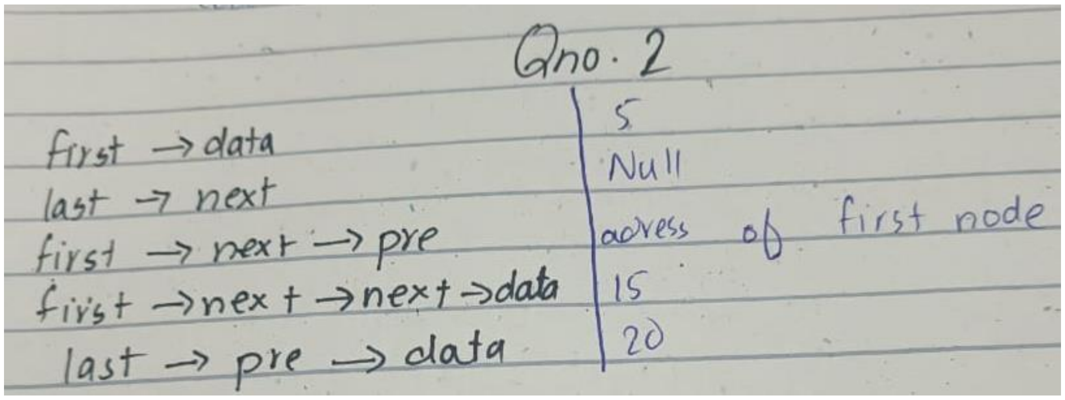
Q no. 1

```
1(a) cout << last ->pre->data
(b) cout << first ->next->data
(c)   while (last -> pre -> data != 9)
        { last = last ->pre
        cout << last -> pre; }
```

Write C++ statements to:

a. Print the value 26 using the pointer ‘last’:

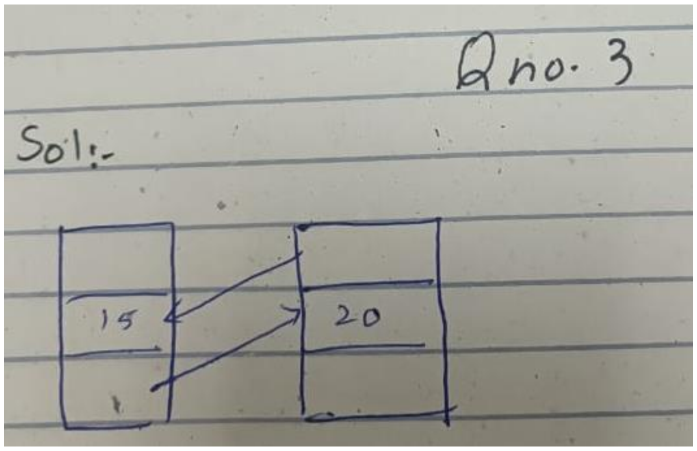
b. Print the value 17 using the pointer ‘first:



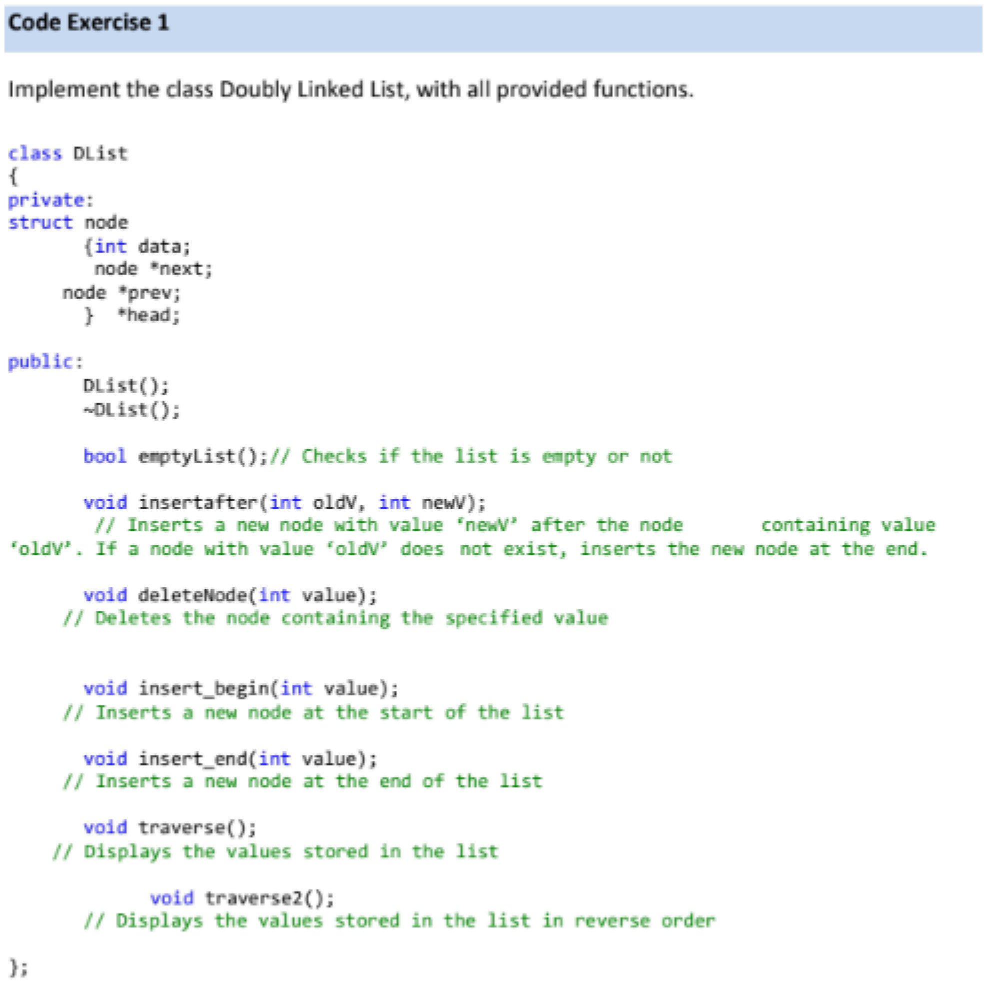
Q no. 2

	5
first -> data	Null
last -> next	adress of first node
first -> next -> pre	15
first -> next -> next -> data	20
last -> pre -> data	

c. Print the address of the node containing value 9 using the pointer ‘last’:



Task #1:



Code Exercise 1

Implement the class Doubly Linked List, with all provided functions.

```
class DList
{
private:
struct node
       {int data;
        node *next;
     node *prev;
       }  *head;

public:
       DList();
       ~DList();

       bool emptyList();// Checks if the list is empty or not

       void insertafter(int oldV, int newV);
        // Inserts a new node with value 'newV' after the node       containing value
'oldV'. If a node with value 'oldV' does not exist, inserts the new node at the end.

       void deleteNode(int value);
     // Deletes the node containing the specified value


       void insert_begin(int value);
     // Inserts a new node at the start of the list

       void insert_end(int value);
     // Inserts a new node at the end of the list

       void traverse();
    // Displays the values stored in the list

             void traverse2();
       // Displays the values stored in the list in reverse order

};
```

Code:

// Online C++ compiler to run C++ program online

```
#include<iostream>
using namespace std;




class DList
{
private:
struct node
{int data;
node *next;
node *prev;
} *head;
public:
DList()
{head=NULL;}

bool emptyList()
{
   if (head == NULL)
       return true;
   else
       return false;
}
void insertafter(int oldV, int newV)
```

```
{
    node *s = head;
    int flag = 0;
    node *temp = new node;
    temp->data = newV;

    if (head == NULL)
    {
        head = temp;
    }
    else if (head->data == oldV && head->next == NULL)
    {
        head->next = temp;
        temp->next = NULL;
    }
    else
    {
        while (s != NULL && s->data != oldV)
        {
            s = s->next;
        }

        if (s != NULL)
        {
            temp->next = s->next;
            temp->prev = s;
            if (s->next != NULL)
            {
                s->next->prev = temp;
            }
            s->next = temp;
            flag++;
        }
    }

    if (flag == 0 && head != NULL)
    {
        s->next = temp;
        temp->prev = s;
        temp->next = NULL;
    }
}
void deleteNode(int value)
{
    int count = 0;
    node *s = head;

    if (head == NULL)
    {
        cout << "linklist is empty" << endl;
```

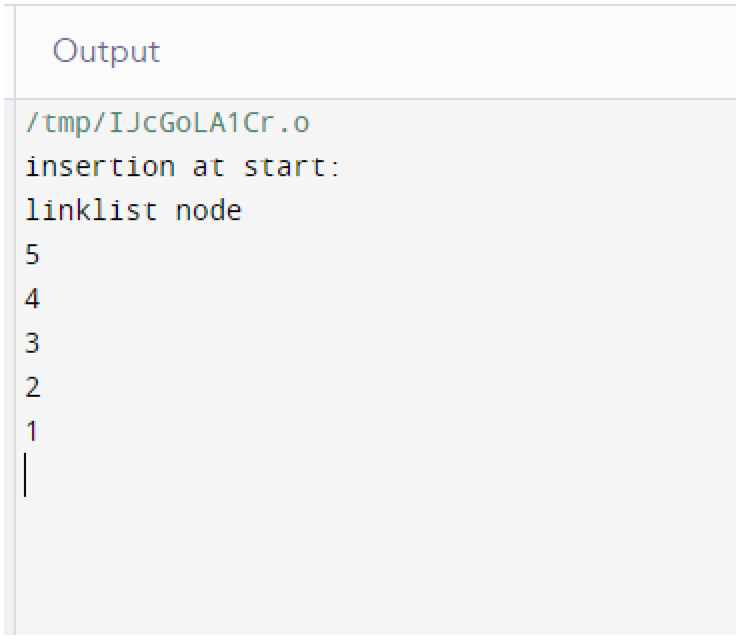
```
}
else if (head->data == value)
{
    node *temp = head;
    head = head->next;
    if (head != NULL)
    {
        head->prev = NULL;
    }
    temp->next = NULL;
    delete temp;
}
else
{
    while (s != NULL)
    {
        if (s->data == value)
        {
            s->prev->next = s->next;
            if (s->next != NULL)
            {
                s->next->prev = s->prev;
            }
            s->next = NULL;
            s->prev = NULL;
```

```
                delete s;
                count++;
                break;
            }
            else
            {
                s = s->next;
            }
        }
    }

    if (count == 0 && s != NULL)
    {
        s->prev = NULL;
        delete s;
    }
}
void insert_begin(int value)
{  node *temp = new node;
temp->data=value;
          if(head==NULL)
          { head=temp;
            head->next=NULL;
            head->prev=NULL;
          }
          else
          { temp->next=head;
            head->prev=temp;
            head=head->prev;
            head->prev=NULL;}
}
void insert_end(int value)
{ node *temp = new node;
temp->data=value;
          if(head==NULL)
          { head=temp;
            head->next=NULL;
          }
          else
          { node*ptr=head;
            while(ptr->next!=NULL)
            { ptr=ptr->next; }
            ptr->next=temp;
            temp->prev=ptr;
            temp->next=NULL;
          }
}
void traverse()
{
          node*s=head;
          while(s!=NULL)
          { cout<<s->data<<endl;
            s=s->next;
          }
}

void traverse2()
{
    if (head == NULL)
    {
        cout << "List is empty" << endl;
        return;
    }

    node *s = head;
    while (s->next != NULL)
    {
        s = s->next;
    }

    while (s != NULL)
    {
        cout << s->data << endl;
        s = s->prev;
    }
```

```
}


};





int main()
{DList d;
cout<<"insertions";
d.insert_begin(1);
d.insert_begin(2);
d.insert_begin(3);
d.insert_begin(4);
d.insert_end(6);
d.insert_end(12);
d.insert_end(10);

d.traverse();
cout<<"traverse2:"<<endl;
d.traverse2();
system("pause");

        return 0;
```

}

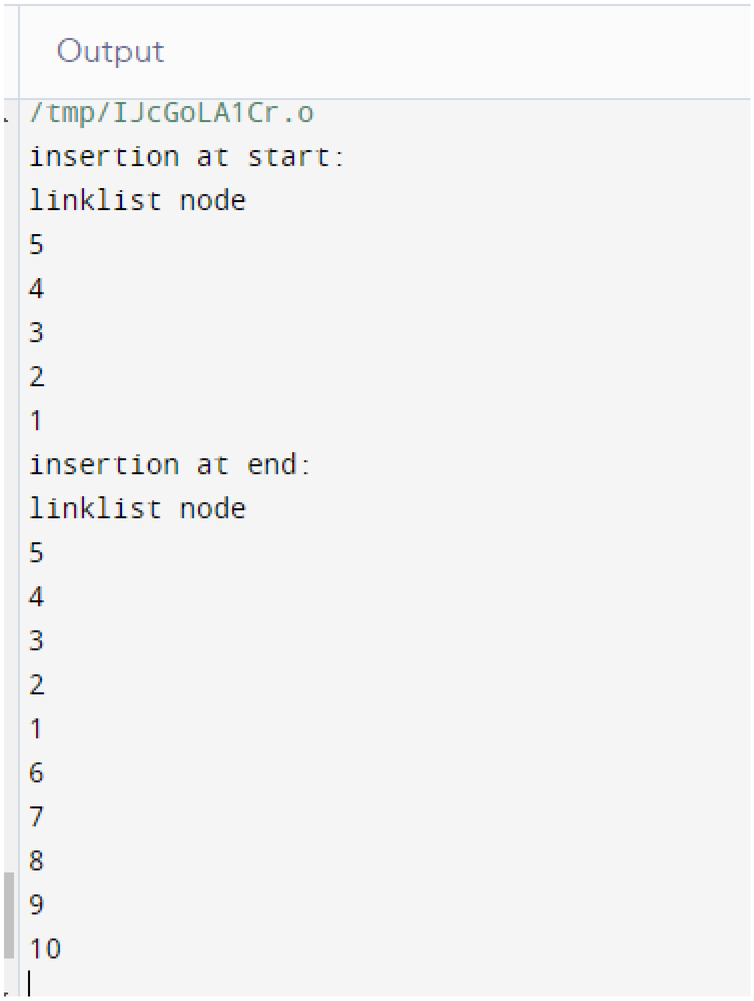
Insertion at start:



Output

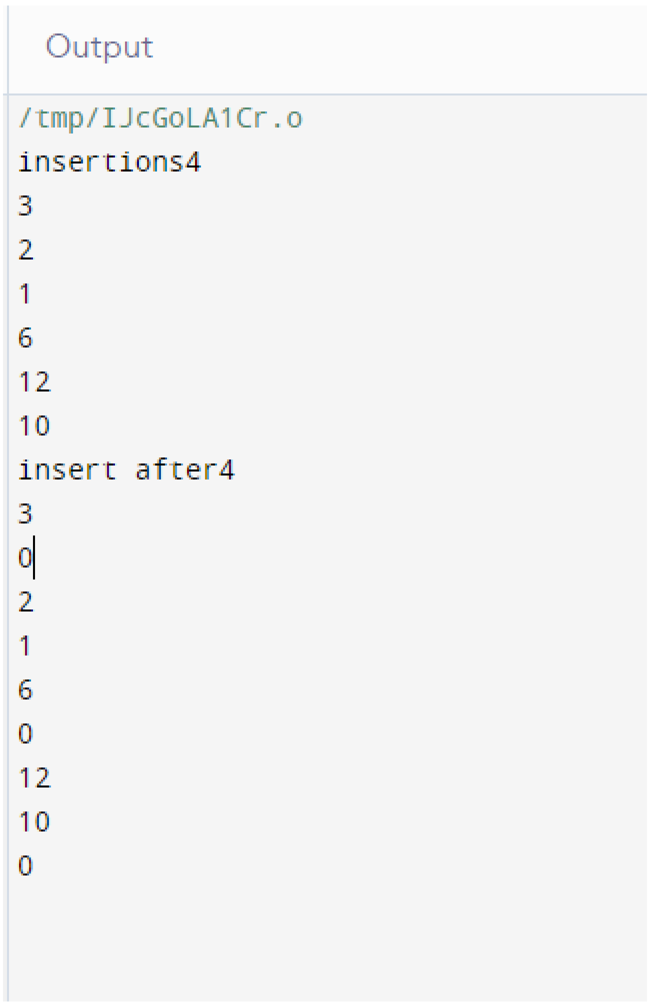
```
/tmp/IJcGoLA1Cr.o
insertion at start:
linklist node
5
4
3
2
1
```

Insertion at end:


Output

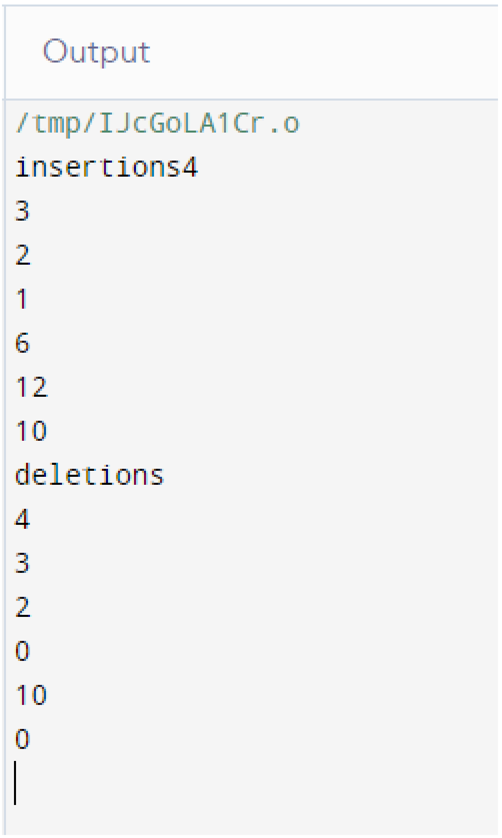
```
/tmp/IJcGoLA1Cr.o
insertion at start:
linklist node
5
4
3
2
1
insertion at end:
linklist node
5
4
3
2
1
6
7
8
9
10
```

Insert after:



```
/tmp/IJcGoLA1Cr.o
insertions4
3
2
1
6
12
10
insert after4
3
0
2
1
6
0
12
10
0
```

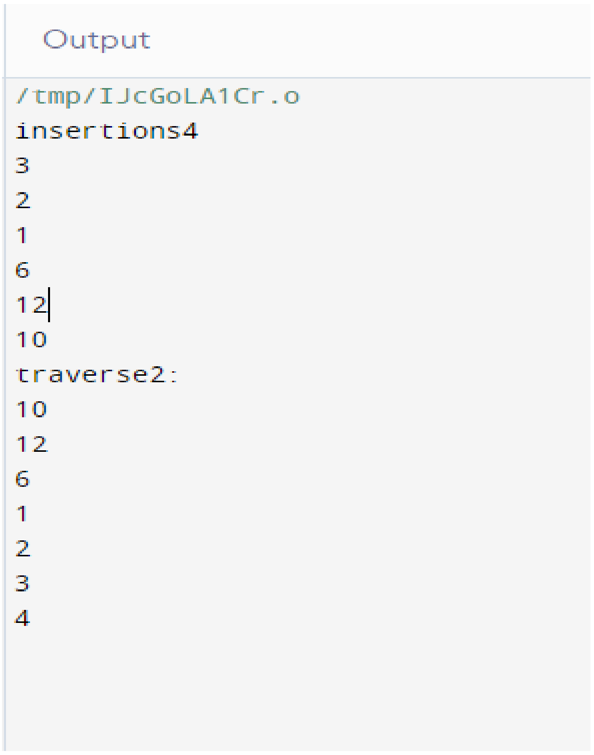
Deletions :



Output

```
/tmp/IJcGoLA1Cr.o
insertions4
3
2
1
6
12
10
deletions
4
3
2
0
10
0
```

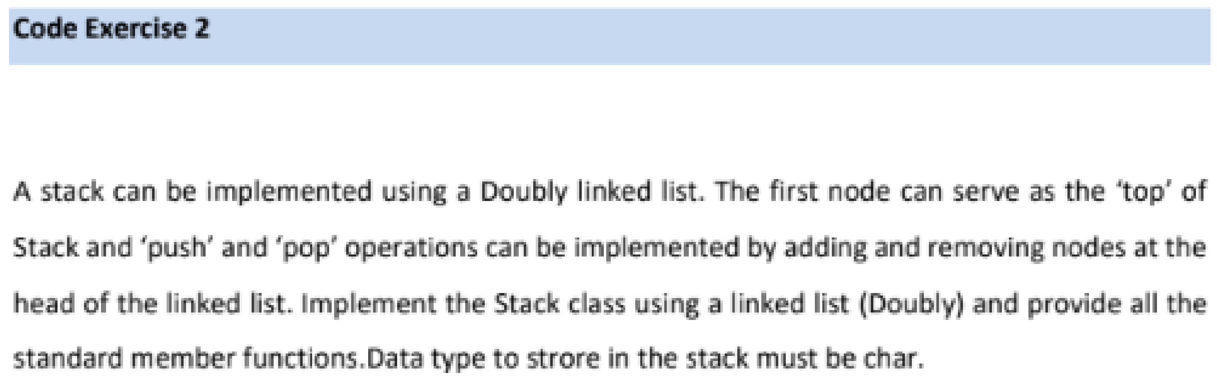
Traverse2:



Output

```
/tmp/IJcGoLA1Cr.o
insertions4
3
2
1
6
12
10
traverse2:
10
12
6
1
2
3
4
```

Task #2:



Code Exercise 2

A stack can be implemented using a Doubly linked list. The first node can serve as the 'top' of Stack and 'push' and 'pop' operations can be implemented by adding and removing nodes at the head of the linked list. Implement the Stack class using a linked list (Doubly) and provide all the standard member functions.Data type to strore in the stack must be char.

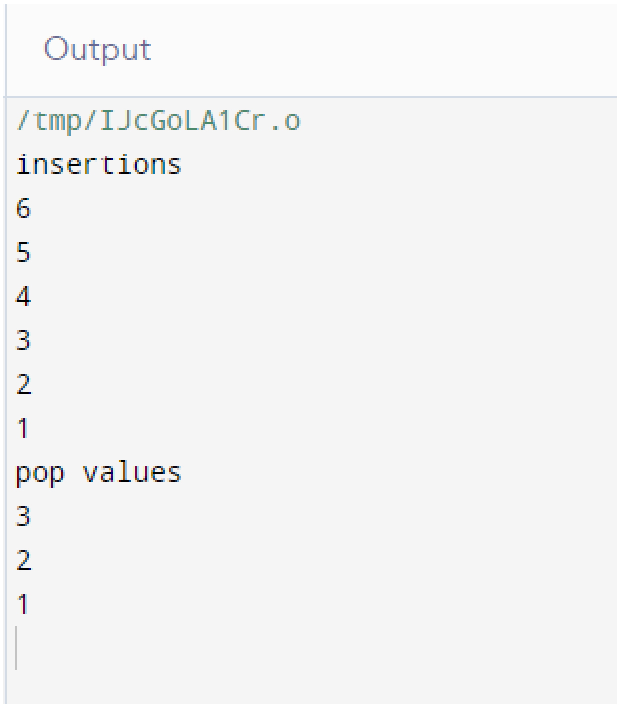
```
// Online C++ compiler to run C++ program online
#include<iostream>
using namespace std;



class DList
{
private:
struct node
{int data;
node *next;
node *prev;
} *top;
```

```
public:
DList()
{top=NULL;}

bool emptyList()
{
    if (top == NULL)
        return true;
    else
        return false;
}

void pop()
{
    if (top == NULL)
    {
        cout << "linklist is empty" << endl;
    }
    else
    {
        node *temp = top;
        top = top->next;
        top->prev = NULL;
        temp->next = NULL;
        delete temp;
    }


}
void push(int value)
{  node *temp = new node;
temp->data=value;
          if(top==NULL)
          { top=temp;
            top->next=NULL;
            top->prev=NULL;
          }
          else
          { temp->next=top;
            top->prev=temp;
            top=top->prev;
            top->prev=NULL;}
}

void traverse()
{
          node*s=top;
          while(s!=NULL)
          { cout<<s->data<<endl;
            s=s->next;
          }
```

```
}


};




int main()
{DList d;
cout<<"insertions"<<endl;
d.push(1);
d.push(2);
d.push(3);
d.push(4);
d.push(5);
d.push(6);
d.traverse();
cout<<"pop values"<<endl;
d.pop();
d.pop();
d.pop();
d.traverse();
system("pause");
```

```
        return 0;
}
```

Output:



```
Output
/tmp/IJcGoLA1Cr.o
insertions
6
5
4
3
2
1
pop values
3
2
1
```

Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

Data structure & Algorithm

Lab # 09

Submitted to:

Sir Rehan

Submitted by:

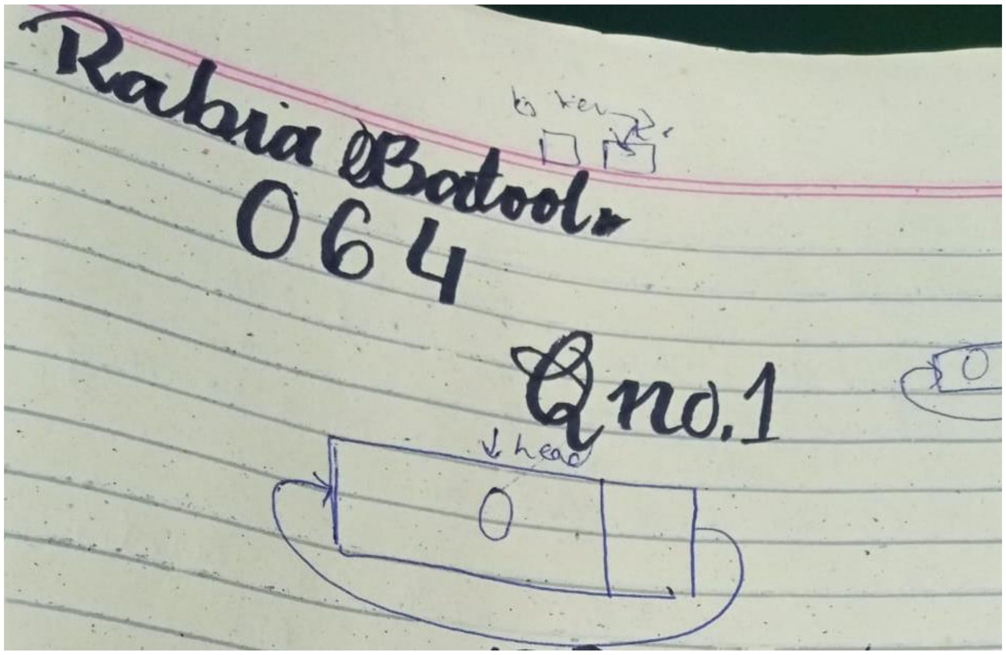
Rabia batool

2022-BSE-064

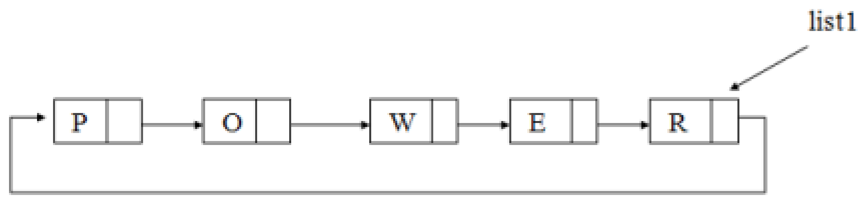
Task 1 :

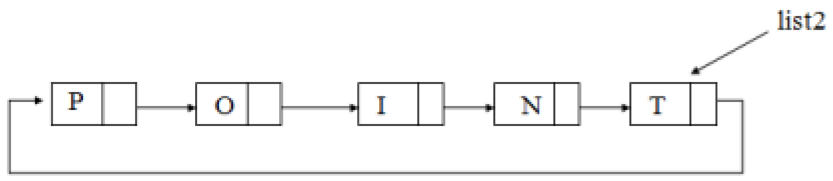
Give answers to the following.

1.Draw a circular linked list of integers with a single node having a value

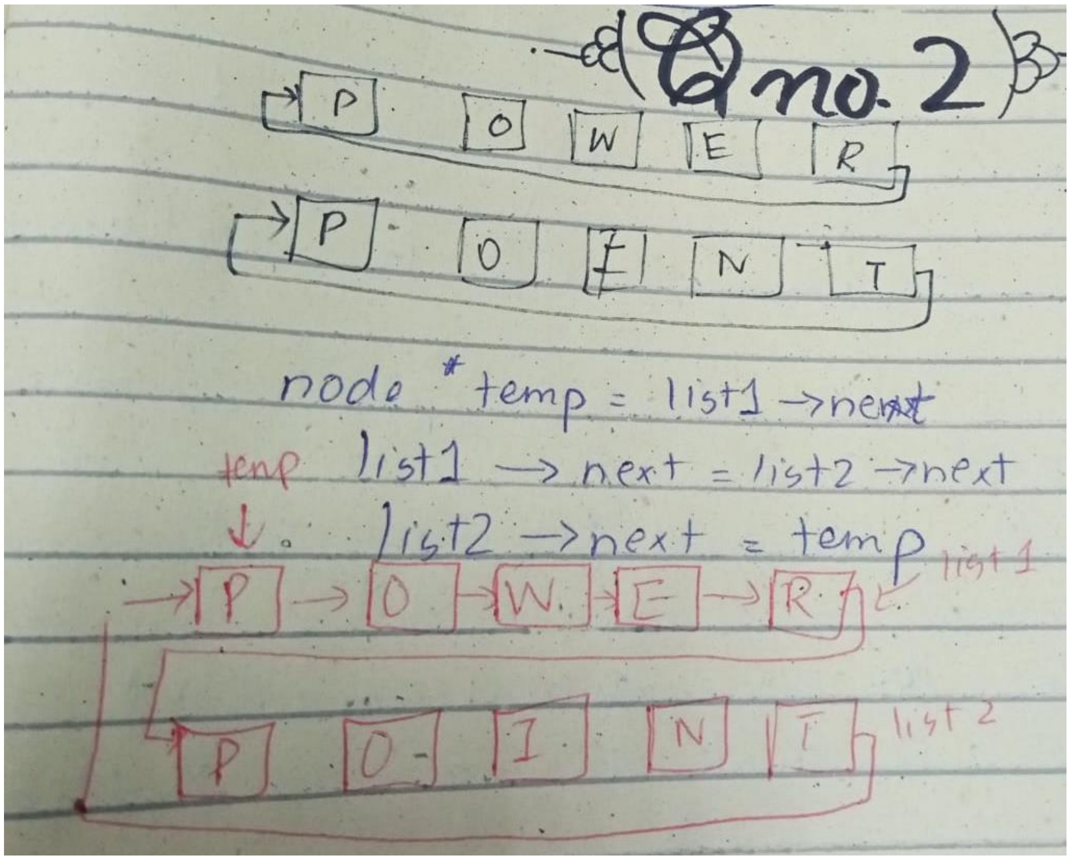


2. Consider the following two circular linked lists with pointers on their last nodes.





Write C++ statements to merge the two lists into one single list with contents POWERPOINT.



```
node *temp = list1 ->next
list1 -> next = list2 ->next
list2 ->next = temp
```

Code task 1

Implement the (class) Circular Linked List to create a list of integers. You need to provide the implementation of the member functions as described in the following.

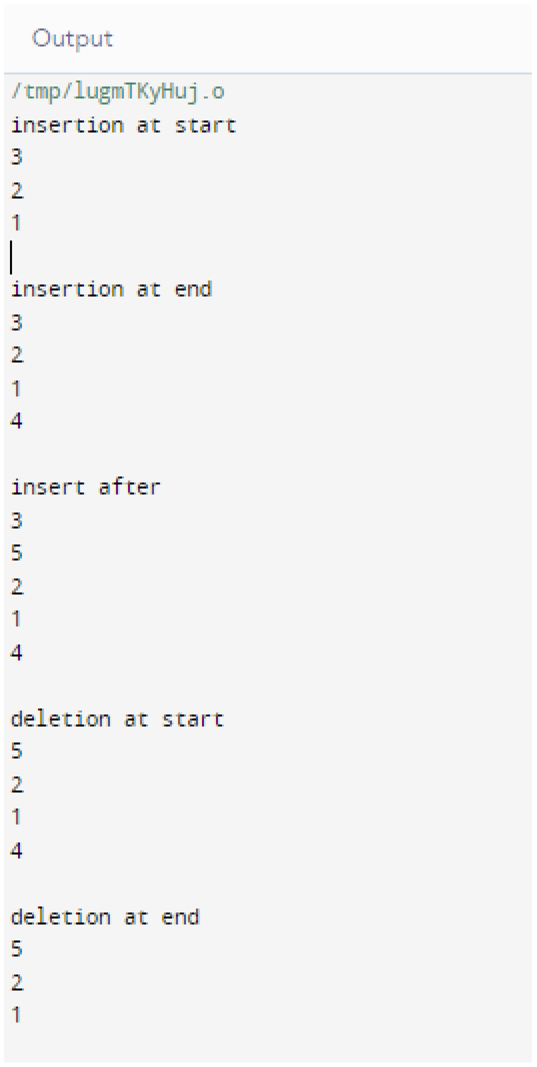
Code:

```
#include<iostream>
```

```
using namespace std;
class circularlist
{public:
struct node
{
int data;
node* next;
}*head;
circularlist()
{head=NULL;
}
bool emptyList()
{
if(head==NULL)
return true;
else
return false;
}

void insert_at_begin(int value)
{
node * ptr1 = new node;
node * temp = head;
ptr1->data = value;
ptr1->next = head;

if (head != NULL)
{
while (temp->next != head)
temp = temp->next;
temp->next = ptr1;
}
else
ptr1->next = ptr1;
head = ptr1;
}
void insert_end(int value)
{
node *temp,*ptr;
ptr=new node;
ptr->data=value;

if(head==NULL){
head=ptr;
ptr->next=head;
}
else{
temp=head;
while(temp->next!=head)
temp=temp->next;
ptr->next=head;
temp->next=ptr;
}
}

void insert_after(int pos,int data) {
node *temp,*t;
temp=head;
```

```
for (int i = 1;i < pos;i++)
{
temp = temp->next;
if (temp == NULL)
{
cout<<"There are less than ";
cout<<pos<<" elements."<<endl;
return;
}
}
t=new node;
t->data=data;
t->next=temp->next;
temp->next=t;
}

void traverse()
{
node *temp = head;
if (head!=NULL)
{
do{
cout<<temp->data<<" "<<endl;
temp = temp->next;
}
while (temp != head);
}
cout<<endl;
}

void del_begin()
{
node *temp,*temp1;
temp=head;
if(head==NULL){
cout<<"\nList has no nodes";
return;
}
if(head->next==head){
head=NULL;
delete temp;
return;
}
temp1=head;
while(temp1->next!=head)
temp1=temp1->next;
head=temp->next;
temp1->next=head;
delete temp;
}
void del_end()
{
node *temp,*t;
temp=head;
if(temp==NULL){
cout<<"Empty linked list ";
return;
}
```

```
if(head->next==head) {
delete temp;
head=NULL; return;
}
t=head->next;
while(t->next!=head){
t=t->next;
temp=temp->next;
}
temp->next=head;
delete t;
}
};
int main()
{
circularlist c1;
c1.insert_at_begin(1);
c1.insert_at_begin(2);
c1.insert_at_begin(3);
c1.traverse();
c1.insert_end(4);
c1.traverse();
c1.insert_after(1,5);
c1.traverse();
c1.del_begin();
c1.del_end();
c1.traverse();
system("pause");
return 0;
}
```



Output

```
/tmp/lugmTKyHuj.o
insertion at start
3
2
1

insertion at end
3
2
1
4

insert after
3
5
2
1
4

deletion at start
5
2
1
4

deletion at end
5
2
1
```

Fatima Jinnah Women University

<u>"lab #10"</u>

<u>COURSE :</u>

DATA STURCTURES

SUBMITTED TO :

SIR RSHAN

SUBMITTED BY :

Rabia Batool (2021-BSE-064)

SECTION :

B

Topic :

Doubly circular linklist

```
// Online C++ compiler to run C++ program online
#include<iostream>
using namespace std;
class CList
{
private:
struct node
{int data;
node *next;
node *pre;
```

```
} *head;
public:
CList(){
head=NULL;
}
bool emptyList(){
if(head==NULL){
return true;
}
else
return false;
}
void insert (int pos, int value){
node *temp=head;
node *t=new node;
t->data=value;
for (int i = 1;i < pos;i++)
{
temp = temp->next;
if (temp == NULL)
{
cout<<"There are less than ";
cout<<pos<<" elements."<<endl;
return;
}
}
t->next=temp->next;
temp->next->pre=t;
temp->next=t;
```

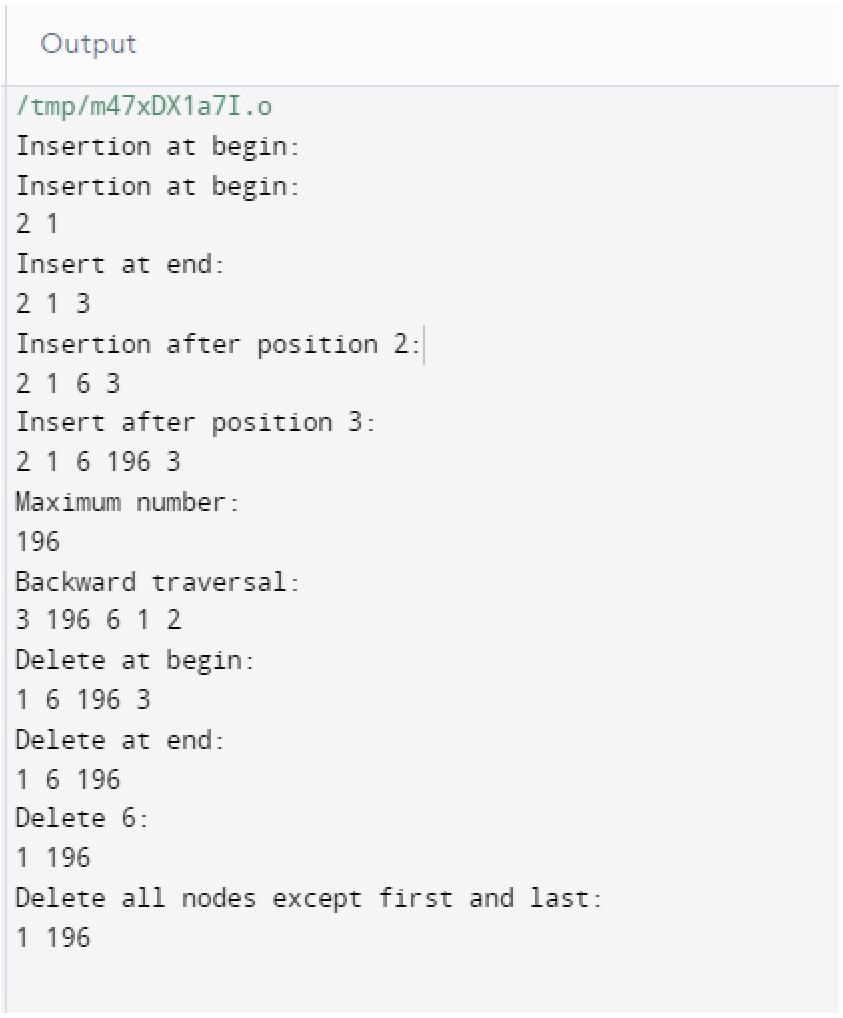
```
t->pre=temp;
}
void insert_begin(int value) {
 node *temp = new node;
 temp->data = value;
 if (head == NULL) {
 head = temp;
 head->next = temp;
 head->pre = temp;
 }
else {
 temp->next = head;
 temp->pre = head->pre;
 head->pre->next = temp;
 head->pre = temp;
 head = temp;
 }
}
void insert_end(int value){
node *temp = new node;
 temp->data = value;
if (head == NULL) {
 head = temp;
 head->next = temp;
 head->pre = temp;
 }
else{
temp->next = head;
 temp->pre = head->pre;
 head->pre->next = temp;
 head->pre = temp;
}
}
void delete_begin(){
node *temp,*temp1;
temp=head;
if(head==NULL){
cout<<"\nList is empty!\n";
return;
}
if(head->next==head){
head=NULL;
delete temp;
return;
}
temp1=head;
while(temp1->next!=head)
temp1=temp1->next;
head=temp->next;
head->pre=temp1;
temp1->next=head;
delete temp;
}
void delete_end(){
node *temp=head;
if (temp == NULL) {
cout << "\n List does not exist";
return;
```

```
}
if (head->next == head) {
delete temp;
head = NULL;
return;
}
else{
node*s=head->pre;
head->pre=s->pre;
s->pre->next=head;
delete s;
}
}
void deletex(int value){
if(head->data==value){
delete_begin();
}
else if(head->pre->data==value){
delete_end();
}
else{
node * temp=head->next;
while(temp->next!=head){
if(temp->data==value){
temp->pre->next=temp->next;
temp->next->pre=temp->pre;
return;
}
else
temp=temp->next;
}
}
}
void traverse() {
 if (head == NULL) {
 cout << "List is empty." << endl;
 return;
 }
 node *s1 = head;
do{
cout<<s1->data<<" ";
s1=s1->next;
}while(s1!=head);
cout<<endl;
}
int max(){
node *max=head;
node *temp=head->next;
do{
if(temp->data>max->data){
max=temp;
}
temp=temp->next;
}while(temp!=head);
return max->data;
}
void two_node_list(){
node *temp2=head->next;
```

```
head->next=head->pre;
head->pre->pre=head;
while(temp2!=head->pre){
node *temp=temp2;
temp2=temp2->next;
delete temp;
}
}
void traverse2(){
if (head == NULL) {
 cout << "List is empty." << endl;
 return;
 }
node *s1 = head->pre;
do{
cout<<s1->data<<" ";
s1=s1->pre;
}while(s1!=head->pre);
cout<<endl;
}
};
int main()
{
CList c1;
cout<<"Insertion at begin:\n";
c1.insert_begin(1);
 cout<<"Insertion at begin:\n";
c1.insert_begin(2);
c1.traverse();
```

```
cout<<"Insert at end:\n";
c1.insert_end(3);
c1.traverse();
cout<<"Insertion after position 2:\n";
c1.insert(2,6);
c1.traverse();
cout<<"Insert after position 3:\n";
c1.insert(3,196);
c1.traverse();
cout<<"Maximum number: \n";
cout<<c1.max()<<endl;
cout<<"Backward traversal: \n";
c1.traverse2();
cout<<"Delete at begin: \n";
c1.delete_begin();
c1.traverse();
cout<<"Delete at end: \n";
c1.delete_end();
c1.traverse();
cout<<"Delete 6: \n";
c1.deletex(6);
c1.traverse();
cout<<"Delete all nodes except first and last: \n";
c1.two_node_list();
c1.traverse();
return 0;
}
```

Output:



```
Output
/tmp/m47xDX1a7I.o
Insertion at begin:
Insertion at begin:
2 1
Insert at end:
2 1 3
Insertion after position 2:
2 1 6 3
Insert after position 3:
2 1 6 196 3
Maximum number:
196
Backward traversal:
3 196 6 1 2
Delete at begin:
1 6 196 3
Delete at end:
1 6 196
Delete 6:
1 196
Delete all nodes except first and last:
1 196
```

Fatima Jinnah Women University

“lab 12”

COURSE :

DATA STURCTURES

SUBMITTED TO :

SIR RSHAN

SUBMITTED BY :

Rabia Batool (2021-BSE-064)

SECTION :

B

Data Structures and Algorithms

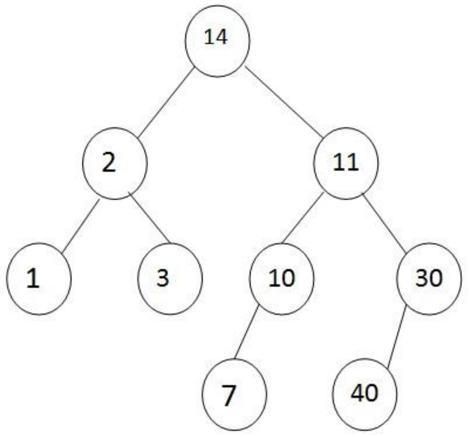
Lab 12

Objective

This lab session is aimed at introducing students to the 'Tree' data structure.

Task 1 :

Give answers to the following.



1. For the given binary tree, state the following.

a. Number of leaf nodes: __________________ ___

b. Number of descendants of node containing 11:___ ________

c. Depth of the tree: ___________

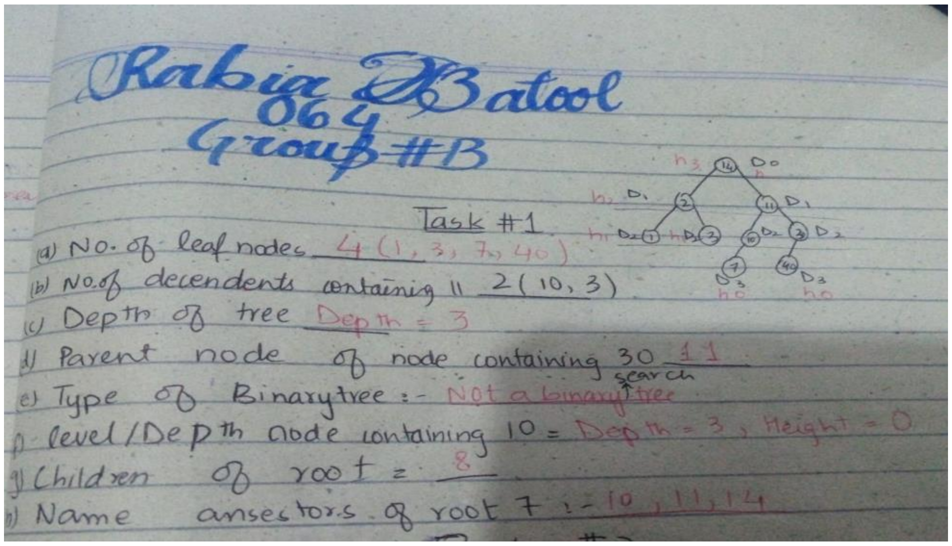
d. Parent node of the node containi ng 30: _____________

e. Type of Binary T ree:______________________________

f. Level/Depth of node containing 10: _________________

g. Children of the root: _____________________

h. Name the ancestors of node containing 7:_________________



Rabia Batool
064
Group #B

Task #1

(a) No. of leaf nodes 4 (1, 3, 7, 40)
(b) No. of decendents containing 11 2 (10, 3)
(c) Depth of tree Depth = 3
(d) Parent node of node containing 30 11
(e) Type of Binary tree :- Not a binary search tree
(f) level / Depth node containing 10 = Depth = 3, Height = 0
(g) Children of root = 8
(h) Name ansestors of root 7 :- 10, 11, 14

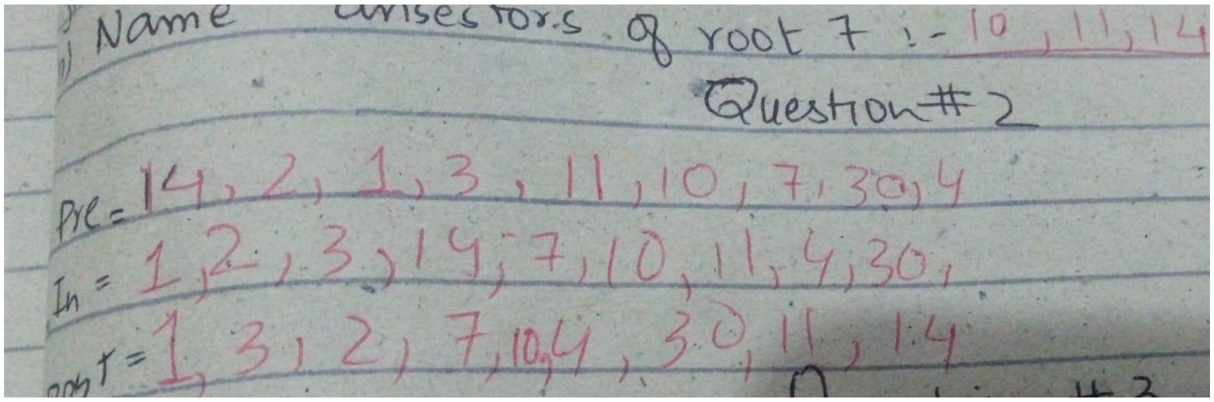
14
2
11
1
3
10
3
7
40
h3
D0
h2
D1
D1
h1
D2
D2
D2
D2
D3
D3
h0
h0

2. Traverse the binary tree given above in pre, post and inorder.

a. Preorder Traversal: ____

b. Post Traversal: ____

c. In-order Traversal: ____



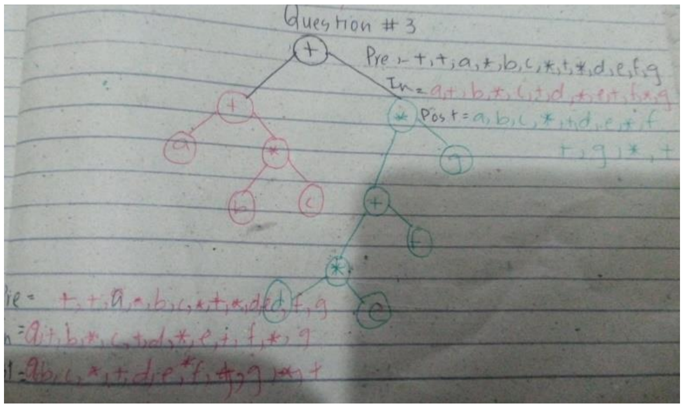
h) Name ansestors of root 7 :- 10, 11, 14

Question # 2

Pre = 14, 2, 1, 3, 11, 10, 7, 30, 4

In = 1, 2, 3, 14, 7, 10, 11, 4, 30,

post = 1, 3, 2, 7, 10, 4, 30, 11, 14

3.	Draw the expression tree of the given algebraic expression and traverse the tree in pre, post and inorder. **(a+b*c)+((d*e+f)*g)** 

Pre :- +,+,a,*,b,c,*,+,*,d,e,f,g
In = a,+,b,*,c,+,d,*,e,+,f,*,g
Post = a,b,c,*,+,d,e,*,f,+,g,*,+

ie = +,+,a,*,b,c,*,+,*,d,e,f,g
n = a,+,b,*,c,+,d,*,e,+,f,*,g
t = a,b,c,*,+,d,e,*,f,+,g,*,+

Code Task

Complete the given class to implement a binary search tree.

```
struct Node
{
	Node *left, *right;
	int data;
	Node()
	{
		left=right=NULL;
	}

};
class bst
{
	Node *root; public:
	bst()
	{
		root=NULL;
	}

	bool isempty();
	void insert(int item);// if already exist do not insert
	bool search(int item);
	void Preorder(node * ptr)
void Postorder(node * ptr)
void Inorder(node * ptr)



};
```

Code :

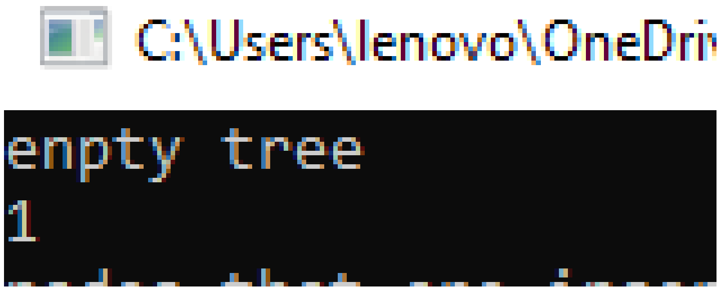
```
// lab 11.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;


struct Node
{
Node *left, *right;
int data;
Node()
{
left=right=NULL;
}
};
class bst
{
Node *root;
```

```
public:
bst()
{
root=NULL;
}
bool isempty()
{if(root==NULL)
return true;
else
       return false;}
void insert(int item)
{Node *s1=root;
Node*s2=0;

while (s1!=0)
{s2=s1;
if(item< s1->data)
       s1=s1->left;
else if(item> s1->data)
       s1=s1->right;
else
       cout<<"valuue already exist\n";
}
Node* temp=new Node;
temp->data=item;
temp->left=0;
temp->right=0;
if(s2==0)
       {root=temp;
cout<<temp->data<<endl;}
else if(item<s2->data)
       {s2->left=temp;
cout<<temp->data<<endl;}
else
       {s2->right=temp;
cout<<temp->data<<endl;}
}

bool search(int item) {
    Node* ptr = root;
    bool found = false;

    while (ptr != NULL && !found) {
        if (item < ptr->data) {
            ptr = ptr->left;
        } else if (item > ptr->data) {
            ptr = ptr->right;
        } else if (item == ptr->data) {
            found = true;
            cout << "Value found: " << ptr->data << endl;
        } else {
            cout << "Unexpected case!" << endl;
        }
    }

    return found;
}
```

```
 void Preorder(Node * ptr)
        {
if(ptr!=NULL)
{
cout << ptr->data<<endl;
Preorder(ptr->left);
Preorder(ptr->right);
}
}

        void Postorder(Node * ptr)
{
if(ptr!=NULL)
{
Postorder(ptr->left);
Postorder(ptr->right);
cout << ptr->data <<endl;
}
}
 void Inorder(Node * ptr)
 {
if(ptr!=NULL)
{
Inorder(ptr->left);
cout << ptr->data<<endl;
Inorder(ptr->right);
}
}
 Node* get()
 {Node *a=root;
        return a;}

};
int _tmain(int argc, _TCHAR* argv[])
{bool a;
bst t;
cout<<"enpty tree"<<endl;
a=t.isempty();
cout<<a<<endl;
cout<<"nodes that are inserted:\n";
t.insert(16);
t.insert(15);
t.insert(19);
t.insert(9);
t.insert(66);
t.insert(8);
t.insert(12);
cout<<"if 1 node is found if zero node is not found"<<endl;
t.insert(61);
cout<<"node that is searched:\n";
cout<<t.search(45)<<endl;
Node*b=t.get();
cout<<"preorder:\n";
t.Preorder(b);
cout<<"ineorder:\n";
t.Inorder(b);
cout<<"prostorder:\n";
t.Postorder(b);
```

```
        system("pause");
        return 0;
}
```

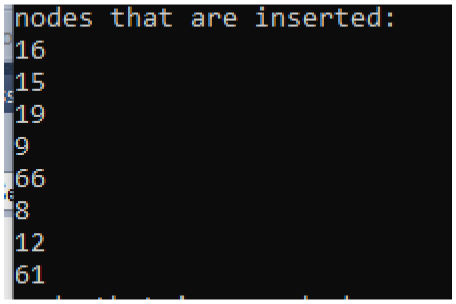
Empty tree:



```
C:\Users\lenovo\OneDri
```

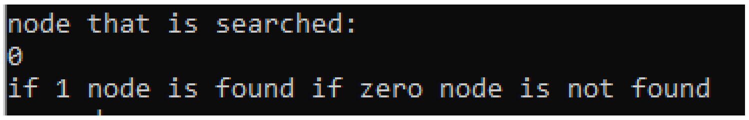
```
enpty tree
1
```

Node to be inserted:



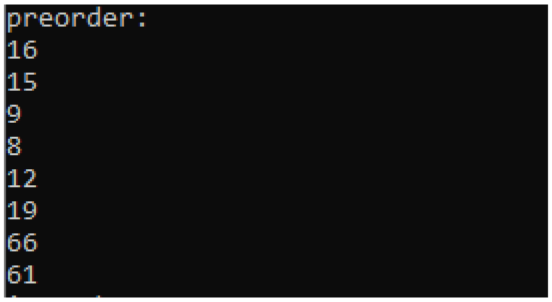
```
nodes that are inserted:
16
15
19
9
66
8
12
61
```

Searched nodes:



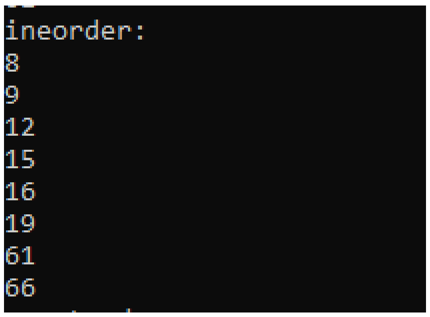
```
node that is searched:
0
if 1 node is found if zero node is not found
```

Pre order:



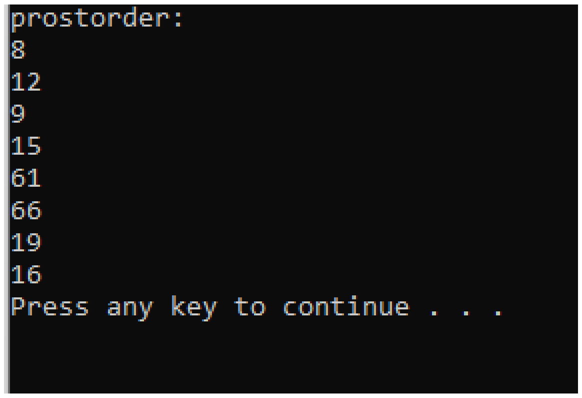
```
preorder:
16
15
9
8
12
19
66
61
```

Inorder :



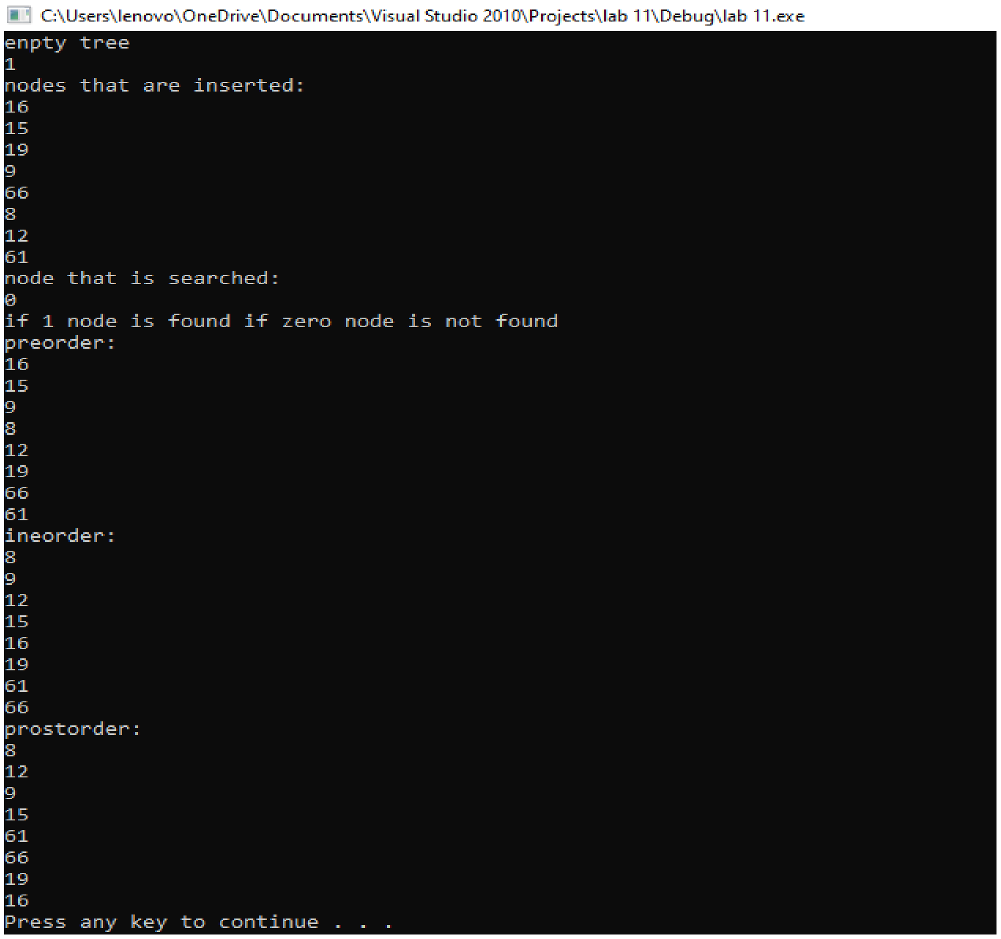
```
ineorder:
8
9
12
15
16
19
61
66
```

Postorder:



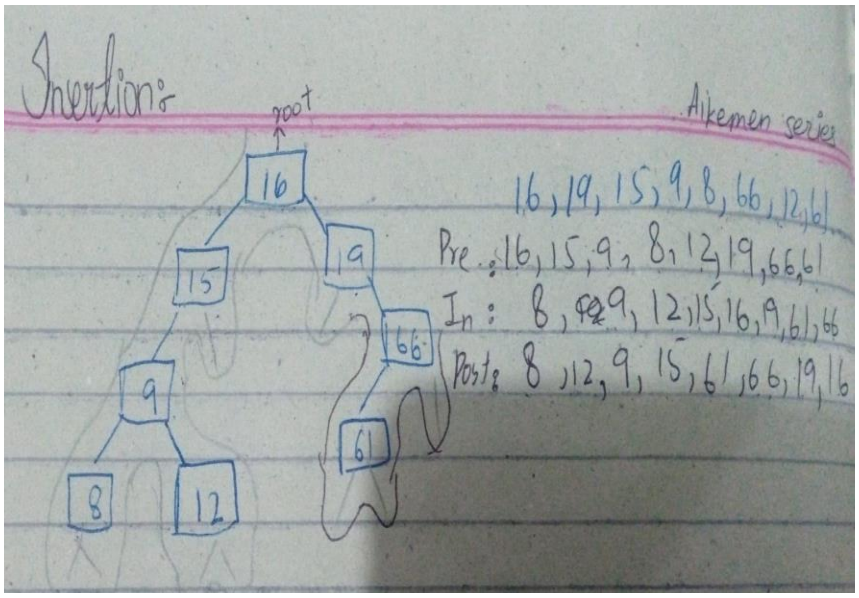
```
prostorder:
8
12
9
15
61
66
19
16
Press any key to continue . . .
```

Output:



C:\Users\lenovo\OneDrive\Documents\Visual Studio 2010\Projects\lab 11\Debug\lab 11.exe

```
enpty tree
1
nodes that are inserted:
16
15
19
9
66
8
12
61
node that is searched:
0
if 1 node is found if zero node is not found
preorder:
16
15
9
8
12
19
66
61
ineorder:
8
9
12
15
16
19
61
66
prostorder:
8
12
9
15
61
66
19
16
Press any key to continue . . .
```



16, 19, 15, 9, 8, 66, 12, 61

Pre: 16, 15, 9, 8, 12, 19, 66, 61

In: 8, 9, 12, 15, 16, 19, 61, 66

Post: 8, 12, 9, 15, 61, 66, 19, 16