LAB 3: STACK

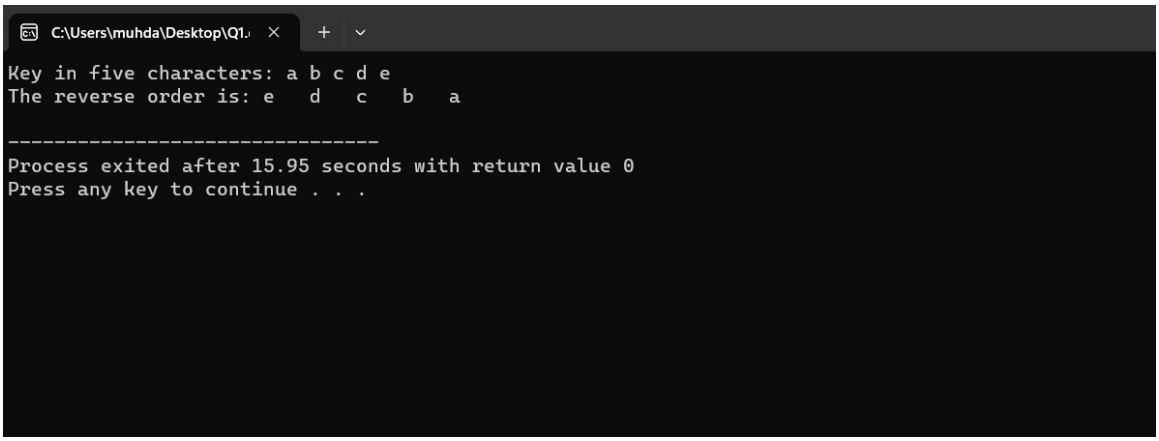
1. Create a program that get 5 characters from user and print these characters in the reverse order. (Modify program in Chapter 3 on Stack)

Sample output:

```
Key in five characters : a b c d e
The reverse order is
e    d     c     b     a
```

ANS:

```cpp
#include <iostream>
//to use stack
#include <stack>
#include <string>

using namespace std;

int main (){

    stack <char> s;
    string input;

    //key in input
    cout << "Key in five characters: ";
    getline (cin,input);

    //loop
    //initialize 1 new char
    for (char c:input){
        //insert stack
        s.push(c);
    }

    cout <<"The reverse order is: ";
    //while loop that continues as long as the s is not empty
    while (!s.empty()){
        cout << s.top() << " ";
        s.pop();
    }

    cout << endl;

    return 0;
}
```


C:\Users\muhda\Desktop\Q1.
Key in five characters: a b c d e
The reverse order is: e d c b a
Process exited after 15.95 seconds with return value 0
Press any key to continue . . .

2. Create a program to get 10 numbers from user and count how many odd numbers and even numbers key in by user. (Modify program in Chapter 3 on Stack)

Sample output :

```
Key in 10 numbers : 10 2 4 5 7 3 5 9 1 8
Numbers entered are : 8 1 9 5 3 7 5 4 2 10
There are 4 even numbers and 6 odd numbers.
```

ANS:

Q2.cpp

```cpp
#include <iostream>
#include <stack>

using namespace std;
int main (){

    stack<int> s;
    int input;
    //even number
    int evenCount = 0;
    //odd number
    int oddCount = 0;

    cout << "Enter 10 numbers: ";
    for (int i = 0; i<10; i++){
        cin >> input;
        s.push(input);
    }

    cout << "Numbers entered are: ";
    //declare another char for stack
    while (!s.empty()){

        int num = s.top();
        cout << num << " ";

        if (num % 2 == 0){
            evenCount++;
        }

        else {
            oddCount++;
```

```
    //even number
    int evenCount = 0;
    //odd number
    int oddCount = 0;

    cout << "Enter 10 numbers: ";
    for (int i = 0; i<10; i++){
        cin >> input;
        s.push(input);
    }

    cout << "Numbers entered are: ";
    //declare another char for stack
    while (!s.empty()){

        int num = s.top();
        cout << num << " ";

        if (num % 2 == 0){
            evenCount++;
        }

        else {
            oddCount++;
        }

        s.pop();
    }

    cout << "\nThere are " << evenCount << " even numbers and " << oddCount << " odd numbers. ";
    return 0;
}
```

```
C:\Users\muhda\Desktop\Q2.
Enter 10 numbers: 10 2 4 5 7 3 5 9 1 8
Numbers entered are: 8 1 9 5 3 7 5 4 2 10
There are 4 even numbers and 6 odd numbers.
--------------------------------
Process exited after 37 seconds with return value 0
Press any key to continue . . .
```

3. Convert the following infix expression to postfix expression.
   i. a+(b*c+d)/e
   ii. (b*b-4*a*c)/(2*a)

Infix	Postfix
a+(b*c+d)/e	abc*d+e/+
(b*b-4*a*c)/(2*a)	bb*4a*c*-2a*/

4. Evaluate the following postfix notation.

   i. 4 5 7 2 + - *

(Tip : Bottom – Top)

+	-	*	

2			
7	9		
5	5	-4	
4	4	4	16

4 * - 4 = -16

ii. 3 4 + 2 * 7 /

+	*	/	
4	2	7	
3	7	14	2

14 / 7 = 2

iii. 5 7 + 6 2 - *

+	-	*	
	2		
7	6	4	
5	12	12	48

12 * 4 = 48

iv. 4 2 3 5 1 - + * *

-	+	*	*	
1				
5	4			
3	3	7		
2	2	2	14	
4	4	4	4	56

14 * 4 = 56

v. 4 2 + 3 5 1 - * +

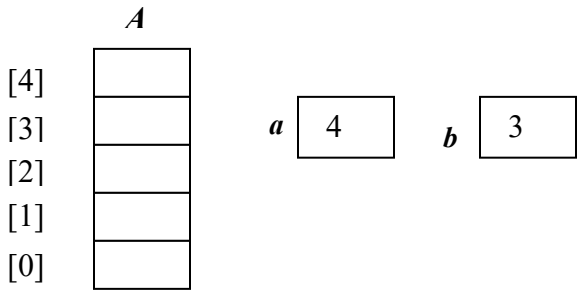
+	-	*	+	
	1			

	5	4		
2	3	3	12	
4	6	6	6	18

12 + 6 = 18

Submission question

Given to you an empty stack, ***A*** of an array ***B [5]***, and two integers ***a*** and ***b***. Draw a sequence diagrams of ***A, a and b*** after each of the following operations:



i. A.push (a*b);

Given A = 4

Given B = 3

[4]			
[3]			
[2]			
[1]	B		
[0]	A	A*B	12

ii. cout<< A.pop ( );

[4]	
[3]	
[2]	
[1]	
[0]	

Output = 12

iii. A.push (a+b);

[4]			
[3]			
[2]			
[1]	B		
[0]	A	A+B	7

iv. A.push(b-a);

[4]			
[3]			
[2]	A		
[1]	B		
[0]	7	B-A	-1

Updated:

[4]	
[3]	
[2]	
[1]	-1
[0]	7

v. a=A.pop( ) / 2;

[4]		
[3]		
[2]		
[1]	-1	-1/2 = 0
[0]	7	7

Updated

[4]	
[3]	
[2]	
[1]	
[0]	7

Value of A is now : 0

vi. A.push(10);

[4]	
[3]	
[2]	
[1]	10
[0]	7

Updated

[4]	
[3]	
[2]	
[1]	10
[0]	7

vii. if (A.pop( ) > 5)

A.push(b);

From Question (iv)

[4]	
[3]	
[2]	
[1]	10
[0]	7

TOS = 10

Since 10 > 5

Thus push (b)

b value = 3

Thus…

[4]	
[3]	
[2]	
[1]	3
[0]	7