**Data Structures and Algorithms**

**Lab Journal - Lab 2**

Name: Saad Ahmad

Enrollment #: 01-134222-130

Class/Section: BS-CS-3A

**Objective**

This lab is intended to introduce students to Stacks and their applications. The students will implement the Stack and employ it in solving the given problems.

**Task 1 :**

Give answers to the following.

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

|  |  |
| --- | --- |
| 2. | Convert the following infix expressions to prefix.  **A\*B +(C/E) – (F+G) :**    **A+(B-D)/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** | **(** | **A** | | **+** | **+** | **A** | | **B** | **+** | **AB** | | **)** |  | **AB+** | | **\*** | **\*** | **AB+** | | **C** | **\*** | **AB+C\*** | | **-** | **-** | **AB+C\*** | | **D** | **-** | **AB+C\*D** | | **+** | **+** | **AB+C\*D** | | **F** | **+** | **AB+C\*DF** | | **\*** | **+** | **AB+C\*DF** | | **G** |  | **AB+C\*DFG\*+-** | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. | 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** | **2** | | **7** | **27** | | **3** | **273** | | **-** | **27-3** | | **/** | **0.5** | | **2** | **0.5 2** | | **1** | **0.5 2 1** | | **5** | **0.5 2 1 5** | | **+** | **0.5 2 6** | | **\*** | **0.5 12** | | **+** | **12.5** |   **Result : 12.5** |

**Task 2 :**

Use the above stack class given in lab manual to write a function to reverse a string.

**For Example:**

alpha->ahpla  
right -> thgir

**Code :**

#include <iostream>

#include <cstring>

using namespace std;

const int MAX = 50;

class Stack {

int top;

char arr[MAX];

public:

Stack() {

top = -1;

}

void push(char a) {

if (top >= MAX) {

cout << "Stack Overflow" << endl;

}

else {

this->arr[++top] = a;

}

}

char pop() {

if (top < 0) {

cout << "Stack Underflow" << endl;

}

else {

return arr[top--];

}

}

bool empty() {

if (top < 0) {

return true;

}

else {

return false;

}

}

void display() {

while (!this->empty()) {

cout << this->pop();

}

}

};

int main() {

Stack s1;

cout << "Enter a String : " << endl;

string arr;

cin >> arr;

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

s1.push(arr[i]);

}

s1.display();

return 0;

}

Output:

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

+++++++++++++++++++++++++