

Lab 7 – Stacks

Implement algorithms using stacks

Task 1. Implement the algorithm to convert an expression from infix to postfix for character operands:

The algorithm to convert from infix expression to postfix expression was discussed in class. The attachment also contains the pseudocode if you would like to *verify* your algorithm. I do not encourage directly using the algorithm without attempting to write it out and understanding it yourself first. Name your source file `myInfixPostfix.cpp`. Assume that you only have operators `+`, `-`, `*`, `/` and that the operands are single characters only. Assume that the expressions are accepted through the standard input.

Test cases you can use:

`A + B - C;`

`(A + B) * C;`

`(A + B) * (C - D);`

`A + ((B + C) * (E - F) - G) / (H - I);`

`A + B * (C + D) - E / F * G + H;`

[LP] Implement the algorithm using the STL stack. The reference for the STL template is [here](#).

[HP] Implement the algorithm using your own implementation of the stack with arrays or linked lists. If you are attempting this approach, you are recommended to test the stack implementation before attempting to use it in the algorithm.

Task 2. Write a program to evaluate a postfix expression:

Include a function in `myInfixPostfix.cpp` with signature `double evaluatePostFix(string expression)` that evaluates an expression provided in postfix format. You may refer to the lecture slides for the algorithm. E.g., `2 10 5 * +` should be evaluated to `52` using your algorithm.

[HP] Implement the algorithm using the STL stack.

[HP] Enable expressions with non-integer and negative numbers.

Specifications

All tasks have components labeled [LP] and [HP]. If you complete ALL the LP components satisfactorily, you will receive a grade of “low pass” on the lab. If you complete ALL the LP components and 2 of the 3 HP components satisfactorily, you will receive a grade of “high pass”. If you do not meet the criteria for a “low pass”, the submission will be marked as “revision needed”.

What to submit:

Your final submission will need to have the files as follows:

- myInfixPostfix.cpp
- Notes to include all your references
- arrayStack.h or linkedListStack.h if you are implementing your own

NOTE: You can look for help on the Internet but refrain from referencing too much. Please cite all your sources in your Notes file.

When to submit:

Submit your lab before **Monday, April 22nd, 11:59pm**. You are strongly advised to submit before Friday, April 12th, 11:59pm.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to provide the reason.

"I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file."