Texas Tech University Whitacre College of Engineering

CS 2413/5401 – Data Structures
Spring 2022
Lab Assignment 4

Acknowledge your collaborators or source of solutions, if any. **Online submission is required.**

While designing your programs or answering items, you are free to come up with your own assumptions based upon concepts and material learned in the course, if every potential specification is not given to you. Just be reasonable and document your assumptions. Such assumptions should not conflict with concepts and material learned in the course.

Your compliance with the "PROGRAMMING STYLE GUIDELINE" for CS 2413/5401 will affect your actual grade. All assignments will be checked for academic misconduct (cheating, plagiarism, collusion, falsifying academic records, misrepresenting facts, violations of published professional ethics/standards, and any act or attempted act designed to give unfair academic advantage to oneself or another student) defined by "OP 34.12: Grading Procedures, Including Academic Integrity" of TTU. Special software will be used to uncover such attempts.

A subset of answers submitted in this lab may be graded.

Objective: Practice using stacks

Tasks:

1. You may work on this assignment by yourself, or you may work with one other student in this course as a team to complete this lab assignment.
    a. Teams larger than 2 people will incur a 25% penalty off the total lab points per extra person.
    b. It is expected that each team member contributes equitably and participates in coding and design ideas.
    c. If a team member is dissatisfied with the performance of the other team member, you are allowed to dissolve the team and continue individually.
        i. Whatever code each team member has contributed may be taken with that team member.
        ii. Try not to let such a decision wait for the day the assignment is due as no extensions will be given if a team is dissolved.
2. Write a C program, problem1.c, to read in a text file, called equations.txt, and determine if the equations are valid.
    a. Example
        i. Text File – equations.txt (may presume that single lowercase letters are used for variables and numbers will only have single digits)
            1. a = ((c+d)*a)/(b/(d-a))
            2. b = 4*[x + 3*(2*x + 1)]
            3. c = –5*{3 – 2*[1 – 4*(3 – 22)]}
            4. d = 5 + 2*{[3 + (2*x – 1) + x] – 2}
            5. e = 5 + 9 * 3
            6. (5 + 9)*3
            7. f (5 + 9)*3
            8. g = 5 + 2*{[3 + (2*x – 1) + x – 2}
            9. h = 5 + 9 * 3)
            10. i = 5 + (9 * 3
        ii. Sample output for each equation where 0 means no error, 1 means variable missing on left side of =, 2 means missing =, and 3 means mismatched ()/[]/{} on the right hand side of the equation. Processing of the equation can stop as soon as an error is found.

1. ***ERROR ID 0 on a = ((c+d)*a)/(b/(d-a))
2. ***ERROR ID 0 on b = 4*[x + 3*(2*x + 1)]
3. ***ERROR ID 0 on c = –5*{3 – 2*[1 – 4*(3 – 22)]}
4. ***ERROR ID 0 on d = 5 + 2*{[3 + (2*x – 1) + x] – 2}
5. ***ERROR ID 0 on e = 5 + 9 * 3
6. ***ERROR ID 1 on (5 + 9)*3
7. ***ERROR ID 2 on f (5 + 9)*3
8. ***ERROR ID 3 on g = 5 + 2*{[3 + (2*x – 1) + x – 2}
9. ***ERROR ID 3 on h = 5 + 9 * 3)
10. ***ERROR ID 3 on i = 5 + (9 *3

b. Use a stack implemented as a linked list to store the opening parentheses, brackets, and braces, in order to determine if error 3 is present in an equation.

- **i.** The stack linked list operations, such as push, pop, and others, should be written as separate functions.
- **ii.** The head of the linked list stack should be a pointer that points to the first node in the stack.
- **iii.** For example, given the equation c = –5*{3 – 2*[1 – 4*(3 – 22)]}, the stack would store the opening (/[/{ as they are encountered when processing the equation. As the closing)/]/} are encountered, the stack would be popped to check for matching ()/[]/{}.
    1. top->{->X
    2. top->{->[->X
    3. top->{->[->(->X
    4. top->{->[->X
    5. top->{->X
    6. top->X

c. The main function should be a driver to call other functions to perform the required tasks to check each equation.

d. No global variables should be used but define macro constants and typedef's may be used.

e. The C library files string.h and ctype.h may be used, but please use the safe string operations, such as *strncmp* in string.h.

f. Report: In the comments at the end of the program, give the following information:

- **i.** Team Member Names
    1. For each team member, detail the work on the program concerning specific work, test cases, and code and functions designed, implemented, and modified, such as
        - **a.** Name
            - **i.** void push (node_t **top, char c); - designed/implemented/modified
            - **ii.** created equations.txt test cases 1, 2, 3, …
            - **iii.** …
    2. A grade penalty of up to 25% of the total assignment points, which will be in addition to any other penalties, may be considered if inequitable contributions are made, not enough detail is present to be convincing, and/or all team members have the same list; i.e., all team members allegedly did the same thing (if both team members do the same thing – one team member is not needed).
- **ii.** Test Cases and Status
    1. Example equations.txt with several equations showing all errors with mismatched), [), (, … – passed/failed
    2. equations.txt with all equations having no errors
    3. equations.txt with all equations having errors of all 3 types
    4. equations.txt with one equation – passed/failed

5. empty equations.txt – passed/failed
6. …

iii. Stack Linked List Analysis

1. Example of a worst-case type of equation that would require many items to be placed on the stack
2. Big O for determining that n equations of max length m are valid
3. Big O of the storage requirements for a stack linked list of size n

Learning Outcomes:

- Understand C program concepts, such as structs and pointers
- Understand how to implement stacks as linked lists

Grading: 50 points

- Standard Deductions - 14 points
- Problem 1 – 36 points (problem1.c, equations.txt, and any other test cases as equations1.txt, equations2.txt, …)
  - Stack Linked List – 12 points
  - Report – 12 points
  - Test Cases – 12 points

Due Date:

2/18/2022, 11:59pm (submitted on Blackboard by ONE team member)