CIS*2520— Assignment #3

Fall 2024

Due: Friday, November 15, 2024@ 23:59

Please submit your assignment solutions as **one zip file** (named as YOUR/UoG/ID_a3.zip, e.g. 1234567_a3.zip) to Dropbox under Assignment 3 before the due date.

You are granted a penalty-free grace period for 48-hours. The grace period ends on Sunday, November 17 23:59. After this time, you cannot submit the assignment. The late assignment (no submission after the grace period) will be marked as ZERO.

If you require a longer grace period (than the default 48 hours) for your assignment, the request must be sent to the course email: cis2520@socs.uoguelph.ca (using your UoG account) BEFORE the due date (Friday November 15, 2024@ 23:59). Please refer to the communication guidelines in week 0 on writing emails.

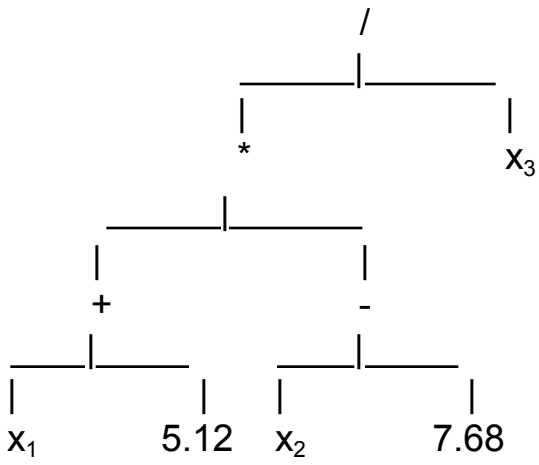
Please refer to the Course Outline and Academic Integrity Video to ensure you understand and comply with the University's Academic Integrity Standards.

1 Binary Tree

Write a C program that takes as input a fully parenthesized, arithmetic expression of binary operators +, −, *, /, and converts the expression into a binary expression tree. Your program should take input from the command line. The entire expression should be in a character string without any space in it.

An input string only includes floating numbers in the format of Y.YY, that is, one digit to the left of the decimal point and two digits to the right of the decimal point, and variables of the form of x_1, x_2, x_3, ...

Your program shall allow for the leaves in the expression tree not only to store floating values but also to store variables of the form x_1, x_2, x_3, ..., For example, expression $(((x_1 + 5.12) * (x_2 - 7.68))/x_3)$ will be converted into a binary expression tree like:



Your program should then show a menu with the following options:

1. Preorder
2. Inorder
3. Postorder
4. Calculate
5. Exit

Description:

(a) If an option of Preorder, Inorder, Postorder is selected, your program should print the expression by the corresponding traversal order (Note: no parentheses for preorder and postorder traversal but fully parenthesized for inorder traversal).

(b) Arithmetic calculation is invoked by option Calculate. Your program should be able to detect an error of divide-by-zero before calculation. You have two ways of implementing this option, and the advanced function gives bonus marks.

- **Calculation Option**: This option performs arithmetic calculations, supporting both expressions with and without variables. The program first checks for any divide-by-zero errors before calculation.
- If the expression contains variables, the program prompts the user to input values for each variable in floating-point format (Y.YY). Once all variable values are entered, they are substituted into the expression.If the variable for any reason is not defined or found you may leave the value as 0.
- The program then evaluates the expression and provides the

answer in 2 decimal places (Y.YY)
- Note there's an error in the calculation such dividing by 0 or unknown operator the calculate function should return and output 0.
- Note: No more than 10 variables will be passed in a given expression

(c) Option Exit terminates your program.

2 Heaps

Write a C program. It reads 200 2-digit integers from text file "f.dat" and stores the integers in an array of 20 rows and 10 columns.

The program treats each row of the array as an object, with the sum of the first three integers being the key, and the other seven integers being the information content.

The program then creates a MAX-HEAP with a node containing an object. You are required to use an array representation of heap, and apply the parental node downheap algorithm in the array representation. The program finally displays the heap as a 20 ×10 array, a row for an object.

3 Assignment 3 Guidelines

- Template files are provided for you to work on this assignment. You **cannot** change the names of the template files, and the names of the functions in the template files.
    - *a3q1_main.c, a3q1_header.h*, and a3*q1_functions.c* are for question 1.
    - *a3q2_main.c, a3q2_header.h*, and a3*q2_functions.c* are for question 1.
- You must create a **readme** file. The information in readme file can be the

same as in Assignment 2. You are welcome to use the previous template and add more info to it as you want, such as the testing input/output from the samples given to you.

- You must create a **makefile** (one single makefile) for compiling the programs. The makefile should be able to generate two executable files.
- The names of your **executables** should be a3q1 and a3q2.
- For this and future assignments, you must compile with "*gcc -Wall -std=c99 -pedantic*". **The code must be tested on the school server before submission**.
- Your program should be able to reasonably detect error conditions (such as invalid input string, wrong .dat files, etc.) and print appropriate error messages.
- **Submission requirement**:
    - Only the following files are required and should be submitted. All the files should be under the folder named YOUR/UoG/ID_a3.
        - *a3q1_main.c, a3q1_header.h*, and a3*q1_functions.c*
        - *a3q1_main.c, a3q1_header.h*, and a3*q1_functions.c*
        - *README*
        - *makefile*
    - No other files should be included in your submission, such as .o files, txt files, etc..
    - Once you have everything, zip the folder and submit it via CourseLink Dropbox.

4 Question-specific Guidelines and Information

Q1: The program for the first question should take input from the command line, same as the second question in Assignment 2. That is, the entire expression should be in a character string without spaces in it.

(a) You will need to write a very simple parser to process the input string. (Parsing a string is to analyze its structure and identify the components.)

(b) You can parse a string by identifying the nested parenthesis pairs, and the three components within each pair.

(c) Within a pair, there is a binary operator, and a left operand and a right operand.

(d) An operand can be a sub-string in a parenthesis pair, a floating number, or a variable.

(e) A pair of parentheses can be identified by counting the opening and closing parentheses included by the pair.

(f) When creating a tree from a string, you may start with the outer most pair of parentheses, identify the operator of it, create the root node for the operator, and then recursively handle the two operands, which may be a sub-string in an inner pairs of parentheses.

(g) To pass an expression as a command line argument, add a \ before each parenthesis.

For example, to pass `(((1.56+4.26)*(x1-2.23))/x2)` to a program, the argument should be `\(\(\(1.56+4.26\)*\(x1-2.23\)\)/x2\)`. Or you can place the expression within single quotation marks, like `./q1 '(((x1+5.12)*(x2-7.68))/x3)'`.

Q2: The 2D array in the second question can be considered as an extension of the 1D array for representing heaps. A file of sample data is provided for your information.

To ensure you receive full marks please fill out all functions in the given function templates.

5 Information on Grading

- q1 is weighted 45%, q2 is weighted 45%, and 10% is for style, comments, documentation, readme, and makefile. For example, in each of the C files, any function should have a brief comment describing its purpose. Also, any section of code where it is not easily apparent what the code does should have a short comment. Don’t forget indentation.

- Any compilation error or warning will result in a mark deduction appropriate to the severity of the error. Memory leaks and memory errors will also result in mark deductions. A maximum of 10% could be applied to deductions.