Coding Assignment 6
ESO207 2024-25-I

October 21, 2024

1 Introduction

In this homework, you will write a C program to implement two algorithms: implementing a calculator with stack and the DP-based matrix chain multiplication

The full marks will be 100. The deadline of this homework is Sunday, October 27, 11 PM IST. Your submissions will not be accepted after this time. Submit a single C file titled 'hw.c' to the Gradescope active assignment titled 'PS1'.

2 Problem Statement

The first line of input will have an option number: 0 for calculator and 1 for matrix multiplication.

2.1 Calculator

The second line of the input will consist of only a number n denoting the total number of numbers and operatosr that are going to be fed into the calculator program. You need to handle an arithmetic expression involving parentheses, the four usual operators with the usual priorities, and $\wedge$ for powering with the highest priority. $-$ and / are left associative i.e. 3-4-5=(3-4)-5=-6 and 12/6/2=(12/6)/2=1; whereas $\wedge$ is right associative i.e $2 \wedge 3 \wedge 2 = 2 \wedge (3 \wedge 2) = 512$. A few sample inputs and outputs are given below in Table 1.

input	**output**
$8 + 3 * 5 \wedge 2 - 9 * 67$	-520
$(8 + (3 * 5)) \wedge 2 - 9 * 67$	-74
$360/5/3 \wedge 1 \wedge 2 + 10 - 5 - 3$	26

Table 1: Sample input-output for calculator

You may assume that all the numbers will be (reasonably small) positive integers only. To simplify parsing, we will change $-$, $+$, *, /, $\wedge$, (,) to respectively -1, -2, -3, -4, -5, -6, -7.

There will a space after each number or operator or parenthesis. For instance, the complete input for the first expression will look as follows:

```
0
11
8 -2 3 -3 5 -5 2 -1 9 -3 67
```

Note that the output simply ends with the final result without any space or newline. You may assume that the input expression is valid and the final answer is an integer.

Remark While all the main ideas of the calculator algorithm using stack was presented in a lecture, there were some errors in the details. It is part of your responsibility to correctly fix those mistakes, resulting in an error-free code.

Remark For the powering ($\wedge$), you may implement it from scratch or you may use the 'pow' function from math.h. Your code will be complied using the 'gcc ... -lm' option that links with the math library.

2.2 Minimum Cost Matrix Multiplication

Input to it will be an integer n followed by n integers denoting the size of the matrices. An example input is given below.

```
1
4
10 5 20 50
```

This input denotes we are multiplying three matrices with dimensions: $10 \times 5, 5 \times 20, 20 \times 50$ respectively. The best option here costs 7500 which starts by multiplying the last pair. Therefore the sample output will be 7500 without any space or newline afterwards.

3 Test Cases

The above 4 test case will be visible and will carry 10 marks each. There will be 3 hidden test cases each of 20 marks. Thus the total will be 100.

4 Submission Instructions

- You must submit a C program titled 'hw.c'. Other programming languages such as C++, Python, etc. are not allowed. Your code must take the input from "stdin" and write the output to "stdout".
- Your code will be automatically graded in Gradescope on some test cases as above which will be hidden from you. Therefore, you must make sure that you understand and precisely follow the expected input-output behavior.
- Please write a single C code and name it as 'hw.c'. This is extremely important. If you violate this, your code will not pass the automatic test cases even if your code runs correctly on your local machine. Common examples of failures include:
  - if you write a C++ program that has the correct input-output behavior
  - if you write two or more different C codes or .h header codes and link them
  - or write a single correct code but name it as 'test.c'

  In any of the above cases, your code will fail. Thus, while you are perfectly allowed to develop your code in your local machine and it works correctly, your code may run into problems in Gradescope until and unless you follow the above instructions.
- Submit your code on *Gradescope* active assignment titled 'PS1'. Otherwise your code will not be graded. In particular, do NOT submit on hello IITK or over email. Email to us (instructor or the TAs), meet us, or start a discussion in helloIITK if you run into any issues.