CS2810 OOAIA: A5

Design Deadline: Feb 10 at 16:40 on Moodle
Code Deadline: Feb 12 at 23:45 on Hackerrank

Contest Link

https://www.hackerrank.com/a5

Objective

- To learn about templates in C++ and implement algorithms to evaluate an arithmetic expression.

Data Structure

Implemented a templated `Stack` class. It should support the standard `push`, `pop` and `top` operations. You are **not allowed** to use the `std::stack` class that C++ STL provides.

Algorithm

Use the templated class to implement the algorithm to evaluate arithmetic expressions in infix form. These expressions consist of terms, operators and parentheses. Terms can be either integers or `Polynomials`. You are free to use code from your previous assignments in your submission.

Input Format

Each testcase will consist of `n` expressions. Each expression has either integer terms or `Polynomial` terms. The first line of the expression is "`int`" or "`poly`" accordingly.

If it is "`int`", the expression will be given in the next line in a single line with space separated tokens, that is, all parentheses, operators and numbers will be space separated.

If the expression is "`poly`", it is followed by an integer on the next line which denotes the number of lines to follow. Each following line is either a polynomial, an operator, or a parenthesis. A polynomial having `m` terms is represented by `2m` numbers in one line, where each pair of numbers represents one term. Each term has the exponent followed by the coefficient.

Please note the following:

- All coefficients are integers. No doubles are given as input.

- The operators we are testing are + (addition), - (subtraction) and * (multiplication).

We have provided an `input.cpp` file on Moodle which provides appropriate logic to take both "`int`" and "`poly`" expressions as input. Feel free to use this in your submissions.

Output Format

The output for an "`int`" expression will be an integer which has to be printed irrespective of whether it is zero or not. The output for a "`poly`" expression will be a `Polynomial` which is printed similar to A3. However as each coefficient is an integer and not a double, there is no need of printing the decimal point or the digits after that. All other rules as in A3 are still followed such as only printing non-zero terms and printing a blank line for a zero polynomial.

Constraints

- There will be a maximum of 100 operations per testcase.
- The maximum degree of any input `Polynomial` will be 10.

Sample Testcase

Input:

```
2                                       → Number of operations
int                                     → Int expression to follow
( ( 2 + 3 ) - 100 ) * 5 - ( 1 - 2 )
poly                                    → Poly expression to follow
11                                      → Next 11 lines define the poly expression
(
(
1 2 3 4                                 → 2x^1 + 4x^3
+
2 3 4 5                                 → 3x^2 + 5x^4
)
-
1 5                                     → 5x^1
)
*
2 3                                     → 3x^2
```

Output:

```
-474
-9x^3 + 9x^4 + 12x^5 + 15x^6
```

Design Submission Format

For the design submission on Moodle, please submit a .tar.gz file named as your roll number.