
CS3460: Data Structures

Lab 01: Stack Applications

Total Points: 20

Problems

IntStack (10 points) **2**

Fun with Chemistry (10 points) **2**

Notes on Grading: Unless otherwise stated, all programs will receive input via `System.in` and will output solutions via `System.out`.

To simplify the grading process, all grading will be automated. When applicable, you will be provided with sample input/output files for testing. You can ensure that your program will receive full marks by testing it with these provided files.

```
$ java YourProgram < input.txt > output.txt
$ diff output.txt correct.txt
```

The first line executes the Java program, redirecting input from a file `input.txt` and writing the output to a file `output.txt`. The second line compares your program's output (now stored in `output.txt`) with the correct answer (stored in `correct.txt`). If these files match exactly, the `diff` program will print nothing. Otherwise, it will list the differences.

Important Note: You are not allowed to use any classes or code from the Java Collections library. While the classes defined in that library would not be an ideal fit for most of our tasks, the purpose of these assignments is to build these data structures from first principles. Programs which import any of these libraries will receive zero points.

Submission: Please submit the files `IntStack.java` and `MolecularMass.java` to AsULearn. Please do not include any other files.

1. **IntStack (10 points):** An integer stack data structure is a stack that holds integers and supports at least the following two operations:

- `push(x)` : adds a new element x to the top of the stack. For the purposes of this assignment, you may assume you will not have more than 100 elements on the stack at once.
- `pop()` : removes the top element of the stack and returns it. If the stack is empty, you should return -1.

You have been provided a file `IntStack.java`. Please complete implementations of the provided methods. You may implement your stack as an array or linked list. You should test your `IntStack` thoroughly with a separate driver program.

2. **Fun with Chemistry (10 points):** A molecule can be represented by a chemical formula consisting of letters and numbers denoting which atoms are present and their quantities. The letter H denotes an atom of hydrogen, a C denotes an atom of carbon, and an O denotes an atom of oxygen. The formula `COOH` would denote a molecule consisting of one carbon, two oxygen, and one hydrogen.

To efficiently express molecules, we use the following rules:

- For atoms that are present in larger quantities, they can be simplified by writing the letter followed by a number to denote repetition. For instance, the formula `COOHHH` can be written as `C02H3` instead. You may assume that when a number is provided, it will be between 2 and 9, inclusive.
- Letters denoting some atoms can be grouped in parenthesis. `CO(OH)` is equivalent to `COOH` and to `C02H`. Atom groups can be followed by numbers to denote that the whole group is repeated. `CH(C02H)3` indicates that the entirety of the `C02H` group is repeated three times.
- Atom groups can be nested within another atom group.

The weight of a molecule is the sum of the masses of all the atoms therein. The weights of each atom you will encounter is listed below:

Atom	Symbol	Weight
Hydrogen	H	1
Carbon	C	12
Oxygen	O	16

Please write a program `MolecularMass.java` that takes a string representation of a molecule as input and prints the appropriate molecular mass. The input will consist of characters C, H, O, parenthesis, and numbers 2-9, strictly following the rules given above (no need to validate input). The input will consist of no more than 100 characters.

Several input/outputs are shown below:

```
Enter the molecule: H2O
The molecular mass of H2O is 18
```

```
Enter the molecule: CH(CO2H)3
The molecular mass of CH(CO2H)3 is 148
```

```
Enter the molecule: ((CH)2(OH2H)(C(H))O)3
The molecular mass of ((CH)2(OH2H)(C(H))O)3 is 222
```