Programming Assignment #1
ADT for Polynomials

Problem Statement

For this programming assignment, we design an abstract data type (ADT) for single-variable polynomials. A polynomial has zero or more terms, each of which consists of an integer coefficient and an integer exponent. This ADT defines methods for addition and subtraction operations on polynomials as well as constructors and other methods.

Requirements & Implementation

- You are to write a Java application that implements the `Polynomial` ADT. A skeleton of the ADT is provided at eTL. The skeleton file is named `PolynomialSkel.java`. You can copy it to your working directory and rename it to `Polynomial.java`. You should then provide implementations of all the methods defined in the class skeleton. You can add more methods as you wish, but are not allowed to remove any method defined in the class skeleton.

- No `main()` method is necessary in your class implementation. We will test and evaluate your class implementation with our own test program (`MainPoly.java`) containing the `main()` method, which is provided at eTL. This test program uses `TextInputStream` class for input and output, which is also provided. So, there is no need to worry about I/O and no need to turn in any additional class containing the `main()` method.

- Once you have compiled your classes and `MainPoly.java` successfully, you can type the following at the Unix shell prompt to run and test your code.

```
% java MainPoly test1
```

  Sample input files (*e.g.*, `test1`) and their output files (*e.g.*, `test1.out`) are provided at eTL.

- Each input file contains two polynomials connected by an operator. Each polynomial is represented by a sequence of (coefficient, exponent) pairs. For example, a polynomial $x^3 + 2x + 4x + 6x^7$ is encoded as `1 3 2 1 4 1 6 7` in the input file. Note that the polynomial terms are not necessarily sorted in any particular order in the input file. However, when a polynomial is printed by the `print` method, the terms must be printed out in decreasing order of exponents. Although two or more terms with the same exponent may appear in an input polynomial, an output polynomial must have only one term for a distinct exponent with a non-zero coefficient.

- For any term in a polynomial, its coefficient and exponent are an integer that can be represented by a 32-bit integer variable. The number of terms in a polynomial can only be limited by the memory capacity of your Java Virtual Machine.

Grading

This assignment is worth 5 percent of your final grade. Efficiency is not considered for grading this assignment. However, if your program does not terminate within a reasonable amount of time (*e.g.*, 100 times the average runtime of the class) and needs to be aborted, it will be considered incorrect. General grading guidelines are:

10 points	Program compiles without errors and is on the right track,
0-50 points	Works on *simple* test cases,
0-40 points	Works on *complex* test cases.

The late submission and regrading policies are described in the course syllabus.

Submission

Turning in your programming assignment must be done at eTL. Do not submit any `.class` or test files. Submit only `.java` files in a single archive (*e.g.*, zip or tar). Refer to the course syllabus for general submission instructions. More specific instructions will be given by the TA.

Due date

The programming assignment is handed out on Monday, Sep. 06, 2021, and due by 11pm on Sunday, Sep. 19, 2021.