CpSc 2120 — Goddard — Fall'17

Lab 2: Arrays and Classes again

Create a class Polynomial that stores polynomials in x with integer coefficients. For example, $2x^6 - 4x^5 + 3x^2 + x$.

The instance variables of your class should be (a) a counter for the number of powers; and (b) an array of Pair's, where a Pair is a struct (or a class) that stores two ints, the power and the coefficient. It is required that **the powers are maintained sorted** in decreasing order at all times. For example, the above polynomial would be stored as:

powerCount = 4

power	6	5	2	1	···
coeff	2	-4	3	1	···

You can assume that no polynomial will ever have more than 100 terms.

The class should be stored in files Polynomial.cpp and Polynomial.h. (The code for Pair can be in its own files, but it is also okay to include it in Polynomial.h. It is also okay to adapt/use the Pair class on the class website as solution to Practice 2.)

Your Polynomial class should have the following methods:

- A constructor that initializes the polynomial to zero
- void incrementBy(int c, int p): increment the current polynomial by $\mathsf{cx^p}$.
  For example,

  ```
  Polynomial A;
  A.incrementBy(3,2);
  A.incrementBy(6,6);
  A.incrementBy(-4,5);
  A.incrementBy(1,1);
  A.incrementBy(-4,6);
  ```

  should produce the above example polynomial.
- A boolean test for whether two polynomials are equal
- an overloaded << operator for output (it's okay if it doesn't have all the bells and whistles; e.g. prints out above polynomial as `2x^6 + -4x^5 + 3x^2 + 1x^1`)

A sample test driver is provided. Adapt as desired. (Do not add main to Polynomial.cpp.)

Submit via handin the files Polynomial.h/cpp (and Pair.h/cpp if created).
(Your driver will not be used in grading.)