Project 2: Subtraction of Polynomials

Problem description

Given two polynomials of the same variable, find their difference. For example,
subtract $x^5 + \quad 2x^3 + 1$ from
$-x^5 + x^4 - 2x^3$,
We get $2x^5 - x^4 + 4x^3 + 1$.

Data structure to represent a polynomial

A polynomial consists of polynomial terms. For example, in $x^5 + 2x^3 + 1$, we have three terms x^5, $2x^3$, and 1. Each term has coefficient, variable, and power. Since we assume that polynomials are of the same variable, without loss of generality, we can name that variable x. So it remains to save the coefficient and power.

```
#ifndef POLY_NODE_H_
#define POLY_NODE_H_
class PolyNode {
public:
    PolyNode(double coeff, int pow);
    double getCoeff() const;
    int getPow() const;
    void setCoeff(double coeff);
    void setPow(int pow);

private:
   double coeff;
   int pow;
};
#endif
```

Next, how do we save all terms of a polynomial? We have two options:

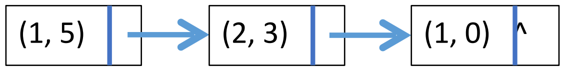
(1) Use an array. Since power of a polynomial term is always non-negative, we can put the coefficient of term x^i in the cell indexed at i. For example, for polynomial $x^5 + 2x^3 + 1$, we can represent it in an array.

Array Index (for power)	0	1	2	3	4	5
Array element (for coefficient)	1	0	0	2	0	1

For example, arr[3] is 2, so we have term $2x^3$.

Array representation is not suitable for this application. Suppose we have a polynomial $x^{100} + 1$, then we need to use an array of size 101 since the indices of an array cannot be omitted, but only two cells in the array are not zero.

(2) Use a linked list of PolyNode* to represent a polynomial. For example, the following list represents $x^5 + 2x^3 + 1$.



In C++, you can use list in its library.
#include <list>
Google "list c++" for operations. Read http://www.cplusplus.com/reference/list/list/.

Some piece of code in main function that may be useful for your program.

```
list<PolyNode*> polyHead;
polyHead.push_back(new PolyNode(2, 3));
polyHead.push_back(new PolyNode(1, 0));
polyHead.push_back(new PolyNode(1, 5));
```

What you need to do

(1) Implement PolyNode.cpp.
(2) Define subtract_polynomials.cpp,
    (2.1) Instantiate two polynomials.
    (2) Write a sortByPow function to sort the terms of a polynomial by power in descending order.
    (3) Define subtract function that takes two polynomials and return the polynomial that is the result of subtracting the second polynomial from the first one.
    (4) Call subtract function on the two polynomials instantiated in (2.1).