

HW5 실습 (과제 아닙니다, 제출하지 않습니다.)

질문: pemds81718@gmail.com

1. 실습 세팅

(a) 다음과 같은 **makefile**을 작성.

```
hw5a: hw5a.o polya.o
    g++ -o hw5a hw5a.o polya.o
hw5a.o polyb.o:polya.h
hw5b: hw5b.o polyb.o
    g++ -o hw5b hw5b.o polyb.o
hw5b.o polyb.o:polyb.h
```

(b) 2개의 다항식을 가진 다음과 같은 입력 파일 **poly.in**을 작성.

(첫 줄은 항의 수가 4이고 각 항이 x^8 과 $-7x^5$ 과 $-x^3$ 과 -3 의 4항으로 이루어진 다항식 이고
둘째 줄은 항의 수가 3이고 각 항이 x^5 과 $2x^3$ 과 -4 인 다항식이다.)

4	1.0	8	-7.0	5	-1.0	3	-3.0	0
3	1.0	5	2.0	3	-4.0	0		

2. 다항식의 구현

(a) 다음 같이 동작하는 프로그램을 작성하려 한다.

```
make hw5a
hw5a < poly.in
```

```
[B6111107@localhost hw2]$ hw2a < poly.in
x^8 -7x^5 -x^3 -3
x^5 +2x^3 -4
x^8 -6x^5 +x^3 -7
```

[주의] 위와 같이 출력 시 지수가 0 이 아닌 항의 경우 그 계수가 1 이면 출력을 생략하고, -1이면 -로 출력되어야 한다.

(b) main 프로그램(hw5a.cpp)을 다음과 같이 작성.

```
#include <iostream>
using namespace std;
#include "polya.h"
int main() {
    Polynomial p1, p2;

    cin >> p1 >> p2; // 2 개의 다항식을 읽어 들인다.
    Polynomial p3 = p1 + p2;
    cout << p1 << p2 << p3;
}
```

(c) 다음 프로그램을 polya.h로 저장.

```
#ifndef POLYNOMIAL_H
#define POLYNOMIAL_H
using namespace std;
class Polynomial; // 전방참조
class Term {
    friend class Polynomial;
    friend ostream& operator<<(ostream&, Polynomial&);
    friend istream& operator>>(istream&, Polynomial&);

private:
    float coef; // coefficient
    int exp;    // exponent
};
class Polynomial {
public:
    Polynomial(); // construct a polynomial  $p(x) = 0$ .
    Polynomial operator+(Polynomial&); // 다항식의 합을 반환
    void NewTerm(const float, const int);
    friend ostream& operator<<(ostream&, Polynomial&);
    friend istream& operator>>(istream&, Polynomial&);

private:
    Term* termArray; // TODO: Array 대신 Linked List 를 사용하는 버전도 구현
    int capacity; // 1로 초기화
    int terms; // 저장된 항의 수로 0으로 초기화
};
#endif
```

(d) 다음 프로그램 polya.cpp를 완성!

```
#include "polya.h"

#include <iostream>
using namespace std;

istream& operator>>(istream& is, Polynomial& p) {
    // #terms and (coefficoent, exponent)의 pair 들을 읽어들인다.
    // 높은차수의 항부터 저장되었다고 가정한다.
    int nofterms;
    float coef;
    int exp;
    is >> nofterms;
    for (int i = 0; i < nofterms; i++) {
        is >> coef >> exp; // 계수와 지수 pair 를 읽어들인다.
        p.NewTerm(coef, exp);
    }
    return is;
}

ostream& operator<<(ostream& os, Polynomial& p) {
    // 출력부분
    return os;
}

Polynomial::Polynomial() : capacity(1), terms(0) {
    termArray = new Term[capacity];
}

void Polynomial::NewTerm(const float theCoeff, const int theExp) {
    // 다항식 뒤에 새로운 항을 추가하는 함수
}

Polynomial Polynomial::operator+(Polynomial& b) {
    // 다항식의 덧셈을 해주는 함수
}
```

3. 다항식의 곱

먼저 poly2.in을 준비

```
3      -1.0 4 3.0 1 -6.0 0
3      2.0 2 1.0 1 2.0 0
```

(a) 다음 같이 다항식 곱하기를 수행하는 프로그램을 작성하려 한다

```
make hw5b
hw5b < poly2.in
```

```
[B6111107@localhost hw2]$ hw2b < poly2.in
-x^4 +3x -6
2x^2 +x +2
-2x^6 -x^5 -2x^4 +6x^3 -9x^2 -12
```

[주의] 위와 같이 출력 시 지수가 0 이 아닌 항의 경우 그 계수가 1 이면 출력을 생략하고, -1이면 -로 출력되어야 한다.

(b) main 프로그램(hw5b.cpp)은 다음과 같다.

```
#include <iostream>
using namespace std;
#include "polyb.h" // include file 을 바꾸었음
int main() {
    Polynomial p1, p2;

    cin >> p1 >> p2; // 2 개의 다항식을 읽어 들인다.
    Polynomial p3 = p1 * p2; // +를 *로 바꾸었음
    cout << p1 << p2 << p3;
}
```

(c) polyb.h 작성. polya.h의 Polynomial 클래스내에서 다음과 같은 public함수 선언을 추가

```
class Polynomial {
public:
    Polynomial(); // construct a polynomial  $p(x) = 0$ .
    Polynomial operator+(Polynomial&); // 다항식의 합을 반환
    void NewTerm(const float, const int);
    friend ostream& operator<<(ostream&, Polynomial&);
    friend istream& operator>>(istream&, Polynomial&);
    Polynomial operator*(Polynomial&);

private:
    Term* termArray; //TODO: Array 대신 Linked List 를 사용하는 버전도 구현
    int capacity; // 1로 초기화
    int terms; // 저장된 항의 수로 0으로 초기화
};
```

(d) polyb.cpp 구현. polya.cpp include file을 polya.h에서 polyb.h로 변경 후, 다음 함수를 추가로 구현해야 함. (힌트: 다항식의 곱 이용)

```
#include "polyb.h"

// ...

Polynomial Polynomial::operator*(Polynomial& b) {
    // 다항식의 곱
}
```