

프로그래밍 과제 04

1. 다음과 같은 방식으로 다항식들을 처리하는 프로그램을 작성하라.

```
$ create f          // 다항식  $f(x) = 0$ 를 정의한다.
$ add f 2 3         //  $f(x)$ 에  $2x^3$ 을 더한다. 즉  $f(x)=2x^3$ 이 된다.
$ add f -1 1        //  $f(x)$ 에  $-x$ 를 더한다. 즉  $f(x)=2x^3-x$ 이 된다.
$ add f 5 0         //  $f(x)=2x^3-x+5$ 가 된다.
$ add f 2 1         //  $f(x)=2x^3-x+5+2x = 2x^3+x+5$ 가 된다.
$ mult f -2         //  $f(x)$ 에 정수  $-2$ 를 곱한다. 즉  $f(x) = -4x^3-2x-10$ 이 된다.
$ calc f 2          //  $f(2)$ 를 계산해서 출력한다.
-46
$ print f           // 차수에 관해 내림차순으로 항들을 정렬하여 다음과 같이 출력한다.
-4x^3-2x-10
$ diff f as h       //  $f$ 를 미분하여  $h$ 라는 이름의 함수로 저장한다. 즉  $h(x) = -12x^2-2$ 
                    // 라는 새로운 다항식을 만들어 저장한다.

$ print h
-12x^2-2
$ calc h 1
-14
$ create g          // 다른 다항함수  $g(x) = 0$ 을 정의한다.
                    // 즉, 이렇게 계속해서 다른 다항식을 정의할수 있어야 한다.
                    // 단, 다항식의 이름이 중복되는 경우 적절한 오류 메시지를 출력한다.
                    //  $g(x) = -4x^3$ 

$ add g -4 3
$ calc g 1
-4
$ add f 1 1         //  $f(x) = -4x^3-x-10$ 
$ calc f 1          // 함수  $f$ 는 계속 유지되어 있어야 한다.
-15
$ print all         // 현재까지 정의된 모든 다항식을 출력한다. 순서는 상관없다.
f(x) = -4x^3-x-10
g(x) = -4x^3
h(x) = -12x^2-2
$ exit              // 프로그램을 종료한다.
```

사용자가 부적절한 명령을 내리는 경우에 대해서는 고려하지 않아도 된다. 다음과 같이 다항식의 각 항을 표현하는 구조체 `Term`과 하나의 다항식을 표현하는 구조체 `Polynomial`을 사용하여 프로그램을 작성하라. 또한 여러 개의 다항식을 저장하기 위한 배열 `polys`를 사용한다. `n`은 정의된 다항식의 개수이다.

```
typedef struct term {          // 하나의 항을 표현하는 구조체
    int coef;                 // 계수
    int expo;                 // 차수
} Term;

typedef struct polynomial {    // 하나의 다항식을 표현하기 위한 구조체
    char name;                // f, g, h등의 다항식의 이름
    int nbr_terms;            // 이 다항식을 구성하는 항의 개수
```

```

    Term terms[MAX_TERMS];          // MAX_TERMS는 100정도로 한다.
} Polynomial;

```

```

Polynomial polys[MAX];              // 여러 개의 다항식을 저장하기위한 배열
int n;                             // 다항식의 개수

```

2. 1번에서 작성한 다항식 프로그램을 다음과 같은 자료구조를 사용하여 동일한 기능을 하도록 수정하라.
즉 구조체 `Polynomial`내의 배열 `terms`와 배열 `polys`를 포인터 타입의 배열로 만들어야 한다.

```

typedef struct term {               // 하나의 항을 표현하는 구조체
    int coef;                      // 계수
    int expo;                      // 차수
} Term;

```

```

typedef struct polynomial {        // 하나의 다항식을 표현하기 위한 구조체
    char name;                    // f, g, h등의 다항식의 이름
    int nbr_terms;                // 이 다항식을 구성하는 항의 개수
    Term *terms[MAX_TERMS];       // MAX_TERMS는 100정도로 한다.
} Polynomial;

```

```

Polynomial *polys[MAX];           // 여러 개의 다항식을 저장하기위한 배열
int n;                           // 다항식의 개수

```