

보고서 #4 (기한: 5/28)

- 다항식 덧셈을 헤더 노드를 사용하지 않고서 해결하고자 한다. 다음과 같이 문제를 해결하라.
 - 3장에서 배운 다항식의 두번째 표현 방식으로 2개의 다항식을 초기화하라.

```
typedef struct { // 항 표현
    float coef;
    integer expo;
} term_type;
```

```
typedef struct { // 다항식 표현
    term_type terms[MAX_TERMS]; // 항의 배열
    int nums; // 항의 개수
} poly;
```

- 다항식의 한 항을 표현하는 타입 nodetype을 정의하라.
- list에 새로운 노드 new를 추가하여 반환하는 알고리즘 insert_last(list, new)를 작성하라.
- 배열로 표현된 다항식을 매개변수로 전달받아서 리스트로 표현된 다항식을 생성하여 반환하는 알고리즘 poly_array2list()를 작성하라. insert_last()를 **이용**하라.

보고서 #4 (계속)

- 두개의 다항식을 표현하는 list1, list2를 전달받아서 다항식 덧셈을 수행하고 그 결과 다항식을 생성하여 반환하는 알고리즘 add_poly_list(list1, list2)를 작성하라. insert_last()를 이용하라.
- 다항식 list를 전달받아서 다음과 같이 출력하는 알고리즘 print_poly(list)를 작성하라: $10x^3 + 3x^2 - 6x + 1$

보고서 #4 (계속)

- 다음과 같이 main()을 작성하고, 테스트하라.

```
main() {  
    poly poly1 = {{{10,5}, {5,2}, {6,1}, {1,0}}, 4}; //  $10x^5 + 5x^2 + 6x + 1$   
    poly poly2 = {{{7,3}, {3,2}, {-6,1}}, 3} //  $7x^3 + 3x^2 - 6x$   
  
    // 리스트로 표현된 다항식을 생성  
    list1 = ploy_array2list(poly1);  
    list2 = poly_array2list(poly2);  
  
    // 다항식 덧셈  
    list3 = add_poly_list(list1, list2);  
  
    // 다항식 출력  
    print_poly(list1); print_poly(list2); print_poly(list3);  
  
    // 다항식 해제  
}
```