

```
1 // polynomial1.cpp : 이 파일에는 'main' 함수가 포함됩니다. 거기서 프로그램 실행이
  시작되고 종료됩니다.
2 //
3 #define _CRT_SECURE_NO_WARNINGS
4 #define MAX_ITEM 100
5 #include <stdio.h>
6
7 int read_polynomial(int);
8 int add_polynomial(int max_degree1, int max_degree2);
9 void print_polynomial(int);
10
11 struct poly_tag {
12     float coef[MAX_ITEM];
13     int degree[MAX_ITEM];
14 }poly1, poly2, poly_result;
15
16 //*****
17 // 다항식 입력 함수, 매개변수는 다항식 순서
18 int read_polynomial(int num) {
19     int poly_item_num=0;
20     float coef;
21     int degree;
22
23     // 다항식의 항의 수 입력
24     while (poly_item_num < 1) {
25         printf("[%d] 번째 다항식의 총 항의 갯수를 입력 하시오: ", num);
26         if (scanf("%d", &poly_item_num) != 1) {
27             printf("scanf error");
28             return -1;
29         }
30     } // while
31
32     // 다항식 계수와 지수 입력
33     for (int no = 0; no < poly_item_num; no++) {
34         scanf("%f %d", &coef, &degree);
35         if (num == 1) {
36             poly1.coef[no] = coef;
37             poly1.degree[no] = degree;
38         }
39         else {
40             poly2.coef[no] = coef;
41             poly2.degree[no] = degree;
42         }
43     } // for
44
45     if (num == 1) {
46         for (int no = 0; no < poly_item_num; no++) {
47             printf("%.2f x^ %d ", poly1.coef[no], poly1.degree[no]);
48         }
49     }
50     else if (num == 2) {
51         for (int no = 0; no < poly_item_num; no++) {
52             printf("%.2f x^ %d ", poly2.coef[no], poly2.degree[no]);
```

```

53     }
54 }
55 printf("Wn");
56 return poly_item_num;
57 }
58 //*****
59 int add_polynomial(int item_num1, int item_num2) {
60     int no=0, no1=0, no2=0 ;
61
62     while (1) {
63         if (poly1.degree[no1] > poly2.degree[no2]) {
64             poly_result.coef[no] = poly1.coef[no1];
65             poly_result.degree[no] = poly1.degree[no1];
66             no1++;
67             if (no1 >= item_num1) { // 2번째 다항식 남은 부분 복사후 루프 종료
68                 while (no2 <= item_num2) {
69                     poly_result.coef[no] = poly2.coef[no2];
70                     poly_result.degree[no] = poly2.degree[no2];
71                     no2++; no++;
72                 }
73                 break;
74             }
75         }
76         else if (poly1.degree[no1] < poly2.degree[no2]) {
77             poly_result.coef[no] = poly2.coef[no2];
78             poly_result.degree[no] = poly2.degree[no2];
79             no2++;
80             if (no2 >= item_num2) { // 1번째 다항식 남은 부분 복사후 루프 종료
81                 while (no1 <= item_num1) {
82                     poly_result.coef[no] = poly1.coef[no1];
83                     poly_result.degree[no] = poly1.degree[no1];
84                     no1++; no++;
85                 }
86                 break;
87             }
88         }
89         else {
90             poly_result.coef[no] = poly1.coef[no1]+ poly2.coef[no2];
91             poly_result.degree[no] = poly1.degree[no1];
92             no1++;
93             no2++;
94             if (no1 >= item_num1) {
95                 while (no2 <= item_num2) {
96                     poly_result.coef[no] = poly2.coef[no2];
97                     poly_result.degree[no] = poly2.degree[no2];
98                     no2++; no++;
99                 }
100                 break;
101             }
102             if (no2 >= item_num2) {
103                 while (no1 <= item_num1) {
104                     poly_result.coef[no] = poly1.coef[no1];
105                     poly_result.degree[no] = poly1.degree[no1];

```

```
106         no1++; no++;
107     }
108     break;
109 }
110 }
111     no++;
112 } // while
113 return no;
114 }
115 //*****
116 void print_polynomial(int item_num) {
117     int no = 0;
118     printf("다항식 덧셈 결과: ");
119     while (no < item_num) {
120         printf("%6.2f x^ %d ", poly_result.coef[no], poly_result.degree[no]);
121         no++;
122     }
123 }
124 //*****
125 int main()
126 {
127     int poly_item_num1 = 0, poly_item_num2 = 0, poly_item_num_result = 0;
128
129     // 다항식 2개 입력
130     poly_item_num1 = read_polynomial(1);
131     poly_item_num2 = read_polynomial(2);
132
133     if (poly_item_num1 > 0 && poly_item_num2 > 0) {
134         poly_item_num_result = add_polynomial(poly_item_num1, poly_item_num2);
135         print_polynomial(poly_item_num_result);
136     }
137 }
138
139
```