

자료구조 다항식 과제

Hints

과제 1 - 버전 1 - 선언부분

```
1 #include <stdio.h>
2
3 #define MAX_DEGREE 100
4
5 typedef struct {
6     int degree;
7     float coefs[MAX_DEGREE+1];
8 } Poly_Type1;
9
10 void read_poly(Poly_Type1 *eq_ptr);
11 void print_poly(Poly_Type1 *eq_ptr);
12
```

과제 1 - 버전 1 - main()

```
13 /*****  
14 int main(void)  
15 {  
16     Poly_Type1 eq_A, eq_B;  
17  
18     read_poly(&eq_A);  
19     printf("A(x) = ");  
20     print_poly(&eq_A);  
21     printf("\n");  
22  
23     read_poly(&eq_B);  
24     printf("B(x) = ");  
25     print_poly(&eq_B);  
26     printf("\n");  
27 }
```

과제 1 - 버전 1 – read_poly() (1)

```
28 /*****  
29 void read_poly(Poly_Type1 *eq_ptr)  
30 {  
31     int i;  
32     int end_of_input = 0;  
33     int expon;  
34     int answer;  
35  
36     eq_ptr->degree = -1;  
37  
38     for (i = 0; i <= MAX_DEGREE; i++)  
39         eq_ptr->coefs[i] = 0.;;  
40
```

과제 1 - 버전 1 – read_poly() (2)

```
41     while (!end_of_input) {
42         printf("Expon : ");
43         scanf("%d", &expon);
44
45         if (expon > eq_ptr->degree)
46             eq_ptr->degree = expon;
47
48         printf("Coef : ");
49         scanf("%f", &eq_ptr->coefs[expon]);
50
51         printf("more input ? (1 or 0) ");
52         scanf("%d", &answer);
53
54         if (answer == 0)
55             end_of_input = 1;
56     }
57
58 }
```

과제 1 - 버전 1 - print_poly()

```
59 /*****  
60 void print_poly(Poly_Type1 *eq_ptr)  
61 {  
62     int i;  
63  
64     for (i = MAX_DEGREE; i >= 0; --i) {  
65         if (eq_ptr->coefs[i] != 0.) {  
66             printf("+ %gX^%d ", eq_ptr->coefs[i], i);  
67         }  
68     }  
69 }
```

과제 1 버전 1의 문제점

- 입력 과정에서의 사용자의 실수에 대한 대비 부족 (MAX_DEGREE를 넘어서는 지수값 입력 등)
- 숫자 입력시 문자를 입력하는 경우 무한 루프에 빠지게 됨
- 출력 과정에서 불필요한 기호(+)나 잘 쓰지 않는 표현들($+1X^3$, $-1X^2$, X^0 등)

과제 1 - 버전 2 - 선언부분

```
1 #include <stdio.h>
2
3 #define MAX_DEGREE 100
4
5 typedef struct {
6     int degree;
7     float coefs[MAX_DEGREE+1];
8 } Poly_Type1;
9
10 int read_poly(Poly_Type1 *eq_ptr);
11 void print_poly(Poly_Type1 *eq_ptr);
12
```


과제 1 - 버전 2 - main()

```
13 /*****  
14 int main(void)  
15 {  
16     Poly_Type1 eq_A, eq_B;  
17  
18     if (read_poly(&eq_A)) {  
19         printf("A(x) = ");  
20         print_poly(&eq_A);  
21         printf("\n");  
22  
23         if (read_poly(&eq_B)) {  
24             printf("B(x) = ");  
25             print_poly(&eq_B);  
26             printf("\n");  
27         }  
28     }  
29 }
```

과제 1 - 버전 2 – read_poly() (1)

```
30 /*****  
31 int read_poly(Poly_Type1 *eq_ptr)  
32 {  
33     int i;  
34     int end_of_input = 0;  
35     int input_error = 0;  
36     int expon;  
37     char answer;  
38  
39     /* 다항식 초기화 */  
40     eq_ptr->degree = -1;  
41  
42     for (i = 0; i <= MAX_DEGREE; i++)  
43         eq_ptr->coefs[i] = 0.0;  
44
```

과제 1 - 버전 2 – read_poly() (2)

```
45     /* 다항식 항 반복 입력 */
46     while (!end_of_input && !input_error) {
47
48         printf("Expon : ");           // 지수부분 입력
49         if ((scanf("%d", &expon) == 1) && (expon <= MAX_DEGREE) && (expon >= 0)) {
50
51             if (expon > eq_ptr->degree) { // 다항식의 차수 처리
52                 eq_ptr->degree = expon;
53             }
54
55             printf("Coef : ");         // 가수 부분 입력
56             if (scanf("%f", &eq_ptr->coefs[expon]) == 1) {
57                 getc(stdin);           // 숫자 입력 과정에서 나온 return 문자 처리
58
59                 printf("more input ? (y or other) ");
60                 scanf("%c", &answer);
61                 if (answer != 'y') {
62                     end_of_input = 1;
63                 }
64             }
65         }
66     }
```

과제 1 - 버전 2 – read_poly() (3)

```
65         else {
66             printf("잘못된 값 또는 입력입니다.\n");
67             input_error = 1;
68         }
69     }
70     else {
71         printf("잘못된 값 또는 입력입니다.\n");
72         input_error = 1;
73     }
74 }
75
76 if (end_of_input) { // 정상적으로 입력이 끝난 경우
77     return 1;
78 }
79 else { // 입력 과정에서 오류가 발생한 경우
80     return 0;
81 }
82 }
```

과제 1 - 버전 2 - print_poly() (1)

```
83 /*****  
84 void print_poly(Poly_Type1 *eq_ptr)  
85 {  
86     int i;  
87     int is_first = 1;  
88  
89     for (i = eq_ptr->degree; i >= 0; --i) {  
90         if (eq_ptr->coefs[i] != 0.) {  
91             if (is_first) {          // 첫 항과 다른 항 구별 목적으로 이용하는 변수  
92                 is_first = 0;  
93             }  
94             else {                  // 첫 항이 아닌 경우 +를 표시, 음수인 경우 음수 부호로 처리  
95                 if (eq_ptr->coefs[i] > 0.)  
96                     printf (" +");  
97                 else  
98                     printf (" ");  
99             }  
100 }
```

과제 1 - 버전 2 - print_poly() (2)

```
101         if (i > 0) {           // X^n 형식으로 출력
102             if (eq_ptr->coefs[i] == 1.0) { // 일반적으로 1이나 -1은 생략
103                 printf("X");
104             }
105             else if (eq_ptr->coefs[i] == -1.0) {
106                 printf("-X");
107             }
108             else {
109                 printf("%gX", eq_ptr->coefs[i]);
110             }
111
112             if (i > 1) {           // X^n에서 n 출력하는 부분
113                 printf ("^%d", i);
114             }
115         }
116         else { /* i == 0 */
117             printf("%g", eq_ptr->coefs[i]);
118         }
119     }
120 }
121 }
```

과제 2 - 선언 부분

```
1 #include <stdio.h>
2 #include <limits.h>
3
4 #define MAX_ELEMENTS 100
5
6 typedef struct {
7     float coef;
8     int exp;
9 } Poly_Element;
10
11 Poly_Element poly_elements[MAX_ELEMENTS];
12
13 int avail_Index = 0;
14
15 typedef struct {
16     int start;
17     int end;
18 } Poly_Type2;
19
20 /*****/
21 int read_poly(Poly_Type2 *eq_ptr);
22 void print_poly(Poly_Type2 *eq_ptr);
23
```

과제 2 – main()

```
24 /*****  
25 int main(void)  
26 {  
27     Poly_Type2 eq_A, eq_B;  
28  
29     if (read_poly(&eq_A)) {  
30         printf("A(x) = ");  
31         print_poly(&eq_A);  
32         printf("\n");  
33  
34         if (read_poly(&eq_B)) {  
35             printf("B(x) = ");  
36             print_poly(&eq_B);  
37             printf("\n");  
38         }  
39     }  
40 }
```


과제 2 – read_poly() (1)

```
41 /*****  
42 int read_poly(Poly_Type2 *eq_ptr)  
43 {  
44     int old_avail_index = avail_Index;    // avail_Index : Global Variable  
45     int min_expon = INT_MAX;  
46     int end_of_input = 0;  
47     int input_error = 0;  
48     int expon;  
49     float coef;  
50     char answer;  
51  
52     eq_ptr->start = avail_Index;  
53  
54     while (!end_of_input && !input_error) {  
55         printf("Expon : ");  
56         if ((scanf("%d", &expon) == 1) && (expon < min_expon) && (expon >= 0)) {  
57             min_expon = expon;    // 내림차순으로 입력받기 위한 변수
```

과제 2 – read_poly() (2)

```
59     printf("Coef : ");
60     if (scanf("%f", &coef) == 1) {
61
62         poly_elements[avail_Index].coef = coef;
63         poly_elements[avail_Index].exp = expon;
64
65         getc(stdin);
66         printf("more input ? (y or other) ");
67         scanf("%c", &answer);
68         if (answer != 'y') {
69             end_of_input = 1;
70         }
71
72         if (avail_Index < MAX_ELEMENTS) {
73             ++avail_Index; // 전역변수 avail_Index 수정
74         }
75         else { // 공간 부족으로 입력 종료
76             end_of_input = 1;
77         }
78     }
```

과제 2 – read_poly() (3)

```
79         else {
80             input_error = 1;
81         }
82     }
83     else if (expon >= min_expon) {
84         printf ("잘못된 지수 값입니다. 반드시 내림차순으로 입력하세요.\n");
85     }
86     else if (expon < 0) {
87         printf ("잘못된 지수 값입니다. 지수는 0 이상이어야 합니다.\n");
88     }
89     else {
90         input_error = 1;
91     }
92 }
93
94 if (end_of_input){ // 정상적으로 입력을 마친 경우 마지막 항 위치 저장
95     eq_ptr->end = avail_Index - 1;
96     return 1;
97 }
98 else { // 입력 과정에서 오류가 발생한 경우 모든 항 취소
99     avail_Index = old_avail_index;
100     return 0;
101 }
102 }
```

과제 2 – print_poly() (1)

```
103 /*****  
104 void print_poly(Poly_Type2 *eq_ptr)  
105 {  
106     int i;  
107     int is_first = 1;  
108  
109     for (i = eq_ptr->start; i <= eq_ptr->end; i++) {  
110  
111         if (i > eq_ptr->start) {  
112             if (poly_elements[i].coef > 0.)  
113                 printf (" +");  
114             else  
115                 printf (" ");  
116         }  
117
```

과제 2 – print_poly() (2)

```
118     if (poly_elements[i].exp > 0) {
119         if (poly_elements[i].coef == 1.0)
120             printf("X");
121         else if (poly_elements[i].coef == -1.0)
122             printf("-X");
123         else
124             printf("%gX", poly_elements[i].coef);
125
126         if (poly_elements[i].exp > 1)
127             printf ("^%d", poly_elements[i].exp);
128     }
129     else {
130         printf("%g", poly_elements[i].coef);
131     }
132
133 }
134 }
```

과제 2의 문제점

- 입력 과정에서의 사용자는 반드시 내림차순으로 항을 입력해야 함
- 자료구조 자체에서 오는 제약으로 현 상태에서는 개선 불가능
- 4장에서 배울 Linked list로 해결