

# REPORT

*Hanseo University*



## 순서리스트와 문자열 함수

**과목:** 자료구조

**학과:** 항공컴퓨터전공

**학년:** 2학년

**학번:** 202200654

**이름:** 김효준

**수강 요일:** 월요일

**제출일:** 2023-04-19

```

1 #include<stdio.h>
2 #define MAX_TERM 1001 /*항의 최대 갯수*/
3
4 typedef struct{
5     double coe; //계수
6     int deg; //지수
7 }Term; //항 구조체
8
9 typedef struct{
10     int n_term; //다항식에서 항의 갯수
11     Term terms[MAX_TERM]; //각 항
12 }Polynomial; //다항식 구조체
13
14 //두 다항식을 더하는 함수
15 Polynomial Add(Polynomial poly1, Polynomial poly2){
16     Polynomial res;
17     int resTop=-1; // == res.n_term - 1
18     int i1, i2=0;
19     //poly1과 poly2의 terms가 deg를 기준으로 오름차순으로 정렬되어 있다고 가정.
20     for(i1=0; i1<poly1.n_term; i1++){
21         while(i2 < poly2.n_term){
22             if(poly1.terms[i1].deg>poly2.terms[i2].deg){
23                 res.terms[++resTop] = poly2.terms[i2];
24             } //만약 poly2의 i2번째 항의 deg가 poly1의 i1번째 항의 deg보다 크다면 빠져나온다.
25             else
26                 break;
27             i2++;
28         } //i2가 poly2의 항의 갯수를 넘어서지 않을 때까지 반복
29         if(poly1.terms[i1].deg != poly2.terms[i2].deg){
30             res.terms[++resTop]=poly1.terms[i1];
31         } //지수가 다르다면 poly1.terms[i1]에 있는 항을 추가
32         else if(poly1.terms[i1].coe + poly2.terms[i2].coe != 0){
33             resTop++;
34             res.terms[resTop]=poly1.terms[i1];
35             res.terms[resTop].coe+=poly2.terms[i2].coe;
36             i2++;
37         } //지수가 같고, 계수의 합이 0이 되지 않는다면 서로 더함
38     }
39     res.n_term = resTop+1;
40     return res;
41 }
42
43 //다항식을 출력하는 함수
44 void printPoly(Polynomial p){
45     printf("%.1f*x^{%d}", p.terms[0].coe, p.terms[0].deg);
46     for(int i=1; i<p.n_term; i++){
47         if(p.terms[i].coe>=0){
48             printf(" + %.1f*x^{%d}", p.terms[i].coe, p.terms[i].deg);
49         }
50         else{
51             printf(" - %.1f*x^{%d}", -p.terms[i].coe, p.terms[i].deg);
52         }
53     }
54     printf("\n");
55 }
56
57 void main(void){
58     //2*x^1000 + 1
59     Polynomial poly1 = {2, {{1.0, 0}, {2.0, 1000}}};
60     printf("poly1: ");
61     printPoly(poly1);
62
63     //x^4 + 10*x^3 + 3*x^2 + 1
64     printf("poly2: ");
65     Polynomial poly2 = {4, {{1.0, 0}, {3.0, 2}, {10.0, 3}, {1.0, 4}}};
66     printPoly(poly2);
67
68     printf("result: ");
69     printPoly(Add(poly1, poly2));
70 }

```

```
poly1: 1.0*x^{0} + 2.0*x^{1000}  
poly2: 1.0*x^{0} + 3.0*x^{2} + 10.0*x^{3} + 1.0*x^{4}  
result: 2.0*x^{0} + 3.0*x^{2} + 10.0*x^{3} + 1.0*x^{4} + 2.0*x^{1000}
```

```

1 #include<stdio.h>
2 #define MAX_TERM 101 /*최대 항의 수+1*/
3
4 typedef struct{
5     int col;
6     int row;
7     int value;
8 } Term; //각 항
9
10 //행렬 a를 전치시켜 res를 생성
11 void transpose(Term a[], Term res[]){
12     //a 항의 갯수가 0개 이하일 경우 에러 처리
13     if(a[0].value<=0){
14         printf("Error: 충분하지 않은 항의 갯수");
15         return;
16     }
17     //res의 행, 열, 값의 갯수를 지정
18     res[0].col = a[0].row;
19     res[0].row = a[0].col;
20     res[0].value = a[0].value;
21
22     //현재 res에 저장한 항의 갯수
23     int cur = 1;
24
25     for(int i=0;i<res[0].col;i++){
26         for(int j=1;j<a[0].value+1;j++){
27             //a[j]의 행이 현재 res의 열이 아닐 경우 a의 다음 항을 탐색
28             if(a[j].row!=i)
29                 continue;
30             //res의 항 저장
31             res[cur].col=a[j].row;
32             res[cur].row=a[j].col;
33             res[cur].value=a[j].value;
34             cur++;
35         }
36     }
37 }
38
39 //행렬을 출력하기 위한 함수
40 void print_matrix(Term a[]){
41     int cur = 1;
42     for(int i=0; i<a[0].col;i++){
43         for(int j=0;j<a[0].row;j++){
44             if(a[cur].col==i && a[cur].row==j){
45                 printf("%3d ", a[cur].value);
46                 cur++;
47             }
48             else{
49                 printf("%3d ", 0);
50             }
51         }
52         printf("\n");
53     }
54 }
55
56 void main(void){
57     Term a[MAX_TERM] = {
58         {6,6,8},
59         {0,0,15},
60         {0,3,22},
61         {0,5,-15},
62         {1,1,11},
63         {1,2,3},
64         {2,3,-6},
65         {4,0,91},
66         {5,2,28}
67     };
68     printf("a\n");
69     print_matrix(a);
70     printf("\n");
71
72     Term b[MAX_TERM];
73     transpose(a, b);
74     printf("b\n");
75     print_matrix(b);
76 }

```

a

15	0	0	22	0	-15
0	11	3	0	0	0
0	0	0	-6	0	0
0	0	0	0	0	0
91	0	0	0	0	0
0	0	28	0	0	0

b

15	0	0	0	91	0
0	11	0	0	0	0
0	3	0	0	0	28
22	0	-6	0	0	0
0	0	0	0	0	0
-15	0	0	0	0	0

```

1 #include<stdio.h>
2 //s의 문자열 길이
3 int strlen(char* s){
4     int len=0;
5     //문자가 '\0'가 나올때까지 len을 계속 더함.
6     while(s[len]!='\0')
7         len++;
8     return len;
9 }
10 //문자열 str1뒤에 str2를 덧붙이기
11 char* strcat(char* str1, char* str2){
12     //str1의 끝부분 부터 str2를 복사함.
13     strcpy(str1+strlen(str1), str2);
14     //문자열 끝에는 \0이 와야 하므로 \0을 집어 넣음
15     *(str1+strlen(str1)+strlen(str2)+1) = '\0';
16     return str1;
17 }
18
19 //문자열을 복사하는 함수
20 char* strcpy(char* str1, char* str2){
21     int len=strlen(str2);
22     //str2의 문자열이 끝날때까지 반복
23     while(*str2!='\0'){
24         //str2의 문자를 str1에 집어넣음
25         *str1 = *str2;
26         //위치 변경
27         str1++;
28         str2++;
29     }
30     *str1='\0';
31     return str1-len;
32 }
33
34 //문자열을 비교하는 함수
35 /*
36 1: str1에서 str2보다 큰 문자가 먼저 나옴
37 0: str1과 str2가 같음.
38 -1: str2에서 str1보다 큰 문자가 먼저 나옴
39 */
40 int strcmp(char* str1, char* str2){
41     while(1){
42         if(*str1>*str2)
43             return 1;
44         else if(*str1<*str2)
45             return -1;
46         if(*str1=='\0' || *str2=='\0')
47             break;
48         str1++;
49         str2++;
50     }
51     return 0;
52 }
53
54 void main(void){
55     char str1[100] = "Hello";
56     char str2[100] = " World!";
57     printf("str1 = %s\nstr2 = %s\n", str1, str2);
58     printf("strlen(str1) = %d\n",strlen(str1));
59     printf("strlen(str2) = %d\n",strlen(str2));
60     printf("strcat(str1, str2) = %s\n", strcat(str1, str2));
61     printf("str1 = %s\nstr2 = %s\n", str1, str2);
62     printf("strcpy(str2, \"Hell\") = %s\n", strcpy(str2, "Hell"));
63     printf("str1 = %s\nstr2 = %s\n", str1, str2);
64     printf("strcmp(str1, str2) = %d\n", strcmp(str1, str2));
65 }

```

```
s1 = Hello
s2 = World!
strlen(s1) = 5
strlen(s2) = 7
strcat(s1, s2) = Hello World!
s1 = Hello World!
s2 = World!
strcpy(s2, "Hell") = Hell
s1 = Hello World!
s2 = Hell
strcmp(s1, s2) = 1
```