

C언어 (CSE2035)

(Chap11. Derived types-enumerated, structure, and union) (1-2)

Ji-Hwan Kim, Ph.D.

Dept. of Computer Science and Engineering

Sogang University

Seoul, Korea

Tel: +82-2-705-8924

Email : kimjihwan@sogang.ac.kr



제출 형식

1. 각 문제에 대한 소스 코드를 압축하여 사이버캠퍼스에 업로드
 - 압축 파일명: "[실습#]학번_이름.zip" (#은 실습번호)
 - 각 소스코드 파일명: "cp실습번호_학번_p문제번호.c"
2. COPY 등의 문제 발생 시 실습 0점 및 각종 불이익을 줄 것

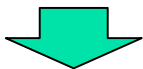


Practice 1.

다음과 같은 구조를 가지는 구조체를 **Vector**로 정의한다.

```
typedef struct{
    float comps;
    int vec_size;
} vector;
```

- 첫번째로 사용자로부터 벡터의 크기를 입력받는다.
- 두번째로 사용자로부터 벡터의 값을 입력받아 벡터를 완성한다.
- 하단의 예제의 경우는 $\vec{v} = (10, 5)$ 이다.



```
int n, m;
vector *v1, *v2;

v1 = (vector*)malloc(sizeof(vector)*n);
v2 = (vector*)malloc(sizeof(vector)*m);
v1[0].comps = 10; v1[1].comps = 5;
...
```



Practice 1.

벡터를 입력 받고, 다음과 같은 함수들을 작성한다.

1. Distance `DistVector (Vector *v1, Vector *v2);`

두 벡터 `v1, v2` 사이의 거리 값을 구해주는 함수이다.

[참고 - 두 점 사이의 거리 공식]

두 개의 벡터 $v1 = \langle x1, y1, z1 \rangle$ 과 $v2 = \langle x2, y2, z2 \rangle$ 의 거리는 다음과 같다.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2 + (z2 - z1)^2}$$

2. Scalar `innerProduct(Vector *v1, Vector *v2);`

두 벡터의 `innerProduct`를 구해주는 함수이다.

[참고 - 두 벡터의 `innerProduct`]

두개의 벡터 $v1 = \langle X1, X2, \dots, Xn \rangle$, $v2 = \langle Y1, Y2, \dots, Yn \rangle$ 의 inner product는 다음과 같다.

$$a \cdot b = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$



Practice 1.

3. Size `vectorSize(Vector *v)`; 벡터의 크기를 구해주는 함수이다.

[참고 – 벡터의 크기]

벡터 $\mathbf{u} = \langle U_x, U_y, U_z \rangle$ 가 주어졌을 때의 크기는 다음과 같다.

$$\|\mathbf{u}\| = \sqrt{u_x^2 + u_y^2 + u_z^2}$$

함수의 원형은 다음과 같다.

```
16  
17 VectorSize vSize(Vector v);  
18 Distance DistVector(Vector v1, Vector v2);  
19 Scalar innerProduct(Vector v1, Vector v2);  
20
```

함수를 구현하기 위해선 **Root**를 사용해야 하기 때문에 `<math.h>`를 이용한다.

컴파일 시 `-lm` 옵션을 사용해야 한다.

Ex) `gcc 14.c -lm`

Practice 1.

입력 : 벡터의 크기 및 원소들 입력(v_1, v_2) 각각 (벡터의 원소개수 10개 이하)

출력 : 벡터의 크기, 벡터의 거리, 벡터의 내적

단, 두개의 벡터의 크기가 같을 경우에만 벡터의 거리, 내적을 계산한다.

Example)

```

sisobusui-MacBook-Pro:15 sisobus$ ./a.out
3
1 2 4
2
2 1
4.58
2.24
v1과 v2의 크기

sisobusui-MacBook-Pro:15 sisobus$ !.
./a.out
4
1 2 3 5
4
2 1 5 7
6.24
8.89 v1과 v2의 크기
3.162 v1과 v2 사이 거리
54.000000 v1과 v2의 내적
  
```

Practice 2.

다항식의 계수와 차수를 입력 받아서 다항식의 덧셈을 출력하는 프로그램을 작성한다.

다음과 같은 구조를 가지는 구조체를 정의한다.

```
typedef struct {
    float coef;
    int expon;
} poly;
```

구현해야 하는 함수 : **poly *poly_sum(int n, int m, poly *a, poly *b)**
크기가 n,m인 다항식 a와 b의 덧셈 결과를 반환하는 함수.

poly a

1.00	2.00
2	3

$x^2 + 2x^3$

result

4.00	6.00	5.00
2	3	7

$4x^2 + 6x^3 + 5x^7$

poly b

3.00	4.00	5.00
2	3	7

$3x^2 + 4x^3 + 5x^7$



Practice 2.

입력 : 다항식 a와 b의 크기 N, M

이후 N개의 줄에 다항식 a의 계수와 차수, M개의 줄에 다항식 b의 계수와 차수가 입력된다. (여기서 계수는 실수가 들어올 수 있다)

출력 : 다항식 a와 b의 합 (출력을 할 때는 아래 예제처럼 낮은 차수부터 출력이 되도록 한다. 또한 상수항이라면 x는 생략하고 출력하도록 한다.)

단, 두 개의 다항식은 항상 오름차순으로 들어온다고 가정.

<주의 사항>

1. 전역변수 사용 불가능.
2. 동적할당을 사용할 것.

```
hinata56@dbpro:~/cprog/prac/14$ ./a.out
2 3
1 3
2 4
3 5
4 6
5 7
1.00x^3+2.00x^4+3.00x^5+4.00x^6+5.00x^7
hinata56@dbpro:~/cprog/prac/14$ ./a.out
2 3
1 2
2 3
3 2
4 3
5 7
4.00x^2+6.00x^3+5.00x^7
```




Practice 3.

다항식의 계수와 차수를 입력 받아서 다항식의 미분 및 적분 결과를 출력하는 프로그램을 작성한다.

다음과 같은 구조를 가지는 구조체를 정의한다.

```
typedef struct {  
    float coef;  
    int expon;  
} poly;
```

구현해야 하는 함수 : **poly *poly_integral(int n, poly *a)**

크기가 n인 다항식 a의 적분 결과를 반환하는 함수.

poly *poly_derivative(int n, poly *a)

크기가 n인 다항식 a의 미분 결과를 반환하는 함수.



Practice 3.

입력 : 다항식 a의 크기 N

이후 N개의 줄에 다항식 a의 계수와 차수가 입력된다. (여기서 계수는 실수가 들어올 수 있다)

출력 : 첫 번째 줄에는 다항식 a의 적분 결과

두 번째 줄에는 다항식 a의 미분 결과

(출력을 할 때는 아래 예제처럼 낮은 차수부터 출력이 되도록 한다. 또한 상수항이라면 x는 생략하고 출력하도록 한다.)

단, 두 개의 다항식은 항상 오름차순으로 들어온다고 가정.

<주의 사항>

1. 전역변수 사용 불가능.
2. 동적할당을 사용할 것.
3. 적분 했을 때의 적분 상수는 1로 고정한다. (예제 그림을 참고할 것)

```
3
1 0
-2 1
1 2
Integral:1.00+1.00x^1-1.00x^2+0.33x^3
Derivative:-2.00+2.00x^1
3
4 2
-6 3
5 7
Integral:1.00+1.33x^3-1.50x^4+0.62x^8
Derivative:8.00x^1-18.00x^2+35.00x^6
```