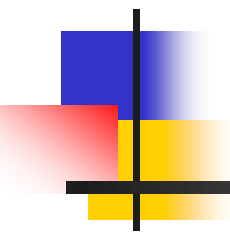


C Programming (CSE2035)

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



Sungwon Jung, Ph.D.

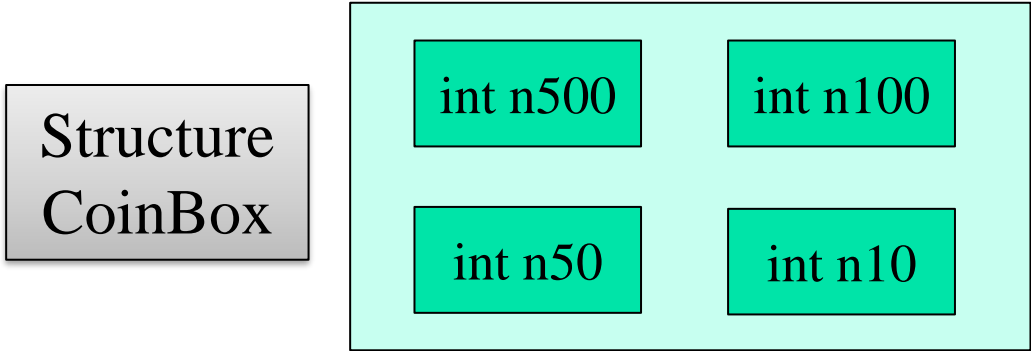
Bigdata Processing & DB LAB
Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea

Tel: +82-2-705-8930

Email : jungsung@sogang.ac.kr

실습 1

자판기에 있는 잔돈 반환 프로그램을 만들어보자. 사용자로부터 거스름돈을 입력 받아서 최소한의 동전을 사용하여 거스름돈에 필요한 동전의 수들을 반환한다. 다음과 같은 구조체를 이용하여 자유롭게 프로그램을 작성한다. 입력되는 거스름돈은 10의 배수로 가정한다.



Example)

```
./a.out
Input money : 660
# of 500 = 1
# of 100 = 1
# of 50 = 1
# of 10 = 1
```

```
./a.out
Input money : 40
# of 500 = 0
# of 100 = 0
# of 50 = 0
# of 10 = 4
```



실습 2

enum도 하나의 정수형 변수로 활용할 수 있다. 즉, **switch**문, 비교연산, **for**, **while**문의 조건 등에 활용 가능하다. (**for**문으로 활용한 경우는 강의 자료 참고) 또한 함수의 인자로도 사용할 수 있다. 날짜를 입력 받아 그 날이 무슨 요일인지 출력해주는 함수를 작성하시오. 함수의 원형은 다음과 같다.

```
enum Date getDate(int year, int month, int day); // 요일에 해당하는 enum값을 return  
char* getDayOfWeek(enum Date currDate); // currDate에 해당하는 요일의 문자열을  
return 한다.
```

이 프로그램의 데이터를 저장하기 위해 다음과 같은 구조체를 구현하시오.

```
struct Calendar{  
    int year, month, day;    // 입력 받은 year, month, day를 저장  
    enum Date date;         // getDate 함수를 통해 얻은 enum Date값을 저장  
    char* dateString;       // getDayOfWeek 함수를 통해 얻은 값을 저장  
};
```

date는 월요일에서 일요일까지의 값을 나타내는 **enum type** 변수이다.

실습 2

주어진 **year, month, day**를 입력하여 요일을 구하는 공식은 다음과 같다.

$$(year + year/4 - year/100 + year/400 + (13 * month + 8) / 5 + day) \% 7$$

이 결과 값(0 ~ 6)과 월요일, 화요일, ..., 일요일 사이의 관계를 구한 뒤 프로그램을 작성한다.

Example)

```
./a.out
Input Year : 2013
Input Month : 11
Input Day : 4
2013-11-04 is Monday
```

```
./a.out
Input Year : 2013
Input Month : 12
Input Day : 25
2013-12-25 is Wednesday
```

Note) 둘 중 편한 방법으로 작성한다. 기능은 동일하다.

```
Calendar cal;
int year;
scanf("%d", &year);
cal.year = year;
```

```
Calendar cal;
scanf("%d", &cal.year);
```

실습 3

다음과 같은 구조를 가지는 구조체를 **Vector**로 정의하고 프로그램을 작성한다.

```
typedef float* Vector_Components;
typedef float Component;

typedef struct{
    Vector_Components comps;
    int vec_size;
}Vector;
```



```
Vector v;
v.vec_size = 2;
v.comps = (Vector_Components)malloc(sizeof(Component)
* v.vec_size);

v.comps[0] = 3; v.comps[1] = 5;
...
```

우선, 사용자로부터 벡터의 크기를 입력 받는다. 그리고 사용자로부터 벡터의 값을 입력 받아 벡터를 완성한다. 위의 경우는 $\vec{v} = (3, 5)$ 를 나타낸다.

사용자로부터 두 개의 벡터를 입력 받고, 다음과 같은 함수들을 작성한다.



실습 3

함수 원형은 다음과 같다.

```
17 VectorSize getSize(Vector v);  
18 Distance getDistance(Vector v1, Vector v2);  
19 Scalar getDotProduct(Vector v1, Vector v2);  
20 Angle getAngle(Vector v1, Vector v2);
```

getSize 함수는 벡터 \vec{v} 의 크기 값을 구해주는 함수이다.

getDistance 함수는 두 벡터 \vec{v}_1 , \vec{v}_2 사이의 거리 값을 구해주는 함수이다.

getDotProduct 함수는 두 벡터 \vec{v}_1 , \vec{v}_2 의 내적 값 $\vec{v}_1 \cdot \vec{v}_2$ 을 구해주는 함수이다.

getAngle 함수는 두 벡터 \vec{v}_1 , \vec{v}_2 가 이루는 각도를 구해주는 함수이다. 각도를 구하기 위해서 다음과 같은 `<math.h>`의 함수를 이용한다.

`double acos(double x);` // x 의 범위는 $-1 \leq x \leq 1$.

실습 3

예를 들어, $\text{acos}(0) = \frac{\pi}{2}$ 이다. 즉, acos 는 **radian**값을 돌려주므로 호도각(°)으로 변환하기 위해서는 $\text{radian} \times 180 / \text{M_PI}$; 의 변환식을 이용한다. Ex) $\text{acos}(0) * 180 / \text{M_PI} = 90(^{\circ})$

여기서 $\text{M_PI} = 3.14$ 라고 가정한다.

단, 두 벡터의 크기가 같을 경우에만 **Distance**, **Dot product**, **Angle**을 출력한다.

Example)

```
./a.out
Input size of Vector v1 : 3
Input 3 components of v1 : 1 0 2
Input size of Vector v2 : 5
Input 5 components of v2 : 1 2 3 4 5

Result
Size of v1 = 2.24
Size of v2 = 7.42
```

```
./a.out
Input size of Vector v1 : 3
Input 3 components of v1 : 2 -2 0
Input size of Vector v2 : 3
Input 3 components of v2 : 4 0 -4

Result
Size of v1 = 2.83
Size of v2 = 5.66
Distance between v1 and v2 = 4.899
Dot Product v1 * v2 = 8
Angle between v1 and v2 = 60.00
```