

Day-20

struct, Bitfield and union

struct

- 구조체는 다양한 변수를 하나의 단위로 통합함

```
struct A{
    int x;
    char y;
};
```

- 구조체 정의 뒤에 하나 이상의 변수를 선언할 수 있음

```
struct A{
    int x;
} a, b;
```

- 열거자는 이름없이 구조체 내에서 선언할 수 있음

```
struct A{
    enum {X, Y}
};

A::X;
```

- 함수 범위와 같은 로컬 범위 내에서 구조체를 선언할 수 있음(일부 제한사항 존재)

```
int f(){
    struct A{
        int x;
    } a;
```

```
    return a.x;
}
```

익명구조체와 무명구조체

- Unnamed struct(무명 구조체)

```
struct {
    int x;
} my_struct; // 무명 구조체
```

- Anonymous struct(익명 구조체)

```
struct S{
    int x;
    struct {int y;}; // 익명 구조체
} // -Wpedantic를 통한 컴파일러경고 발생
```

Bitfield

- 비트필드는 미리 정의된 비트 너비를 가진 구조의 변수
- 비트필드는 바이트대신 비트를 저장할 수 있음

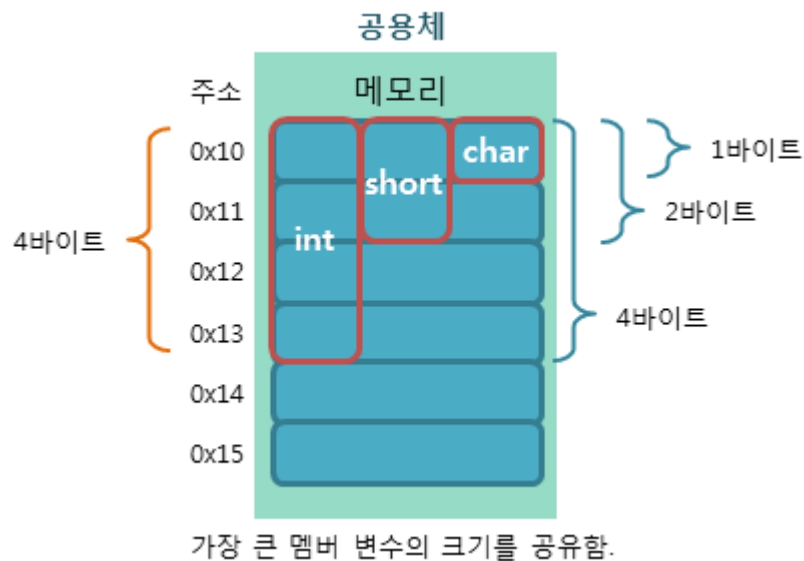
```
struct S1{
    int b1 : 10; // range [0, 1023] 2^10 만큼
    int b2 : 8;  // range [0, 255] 2^8 만큼
}; // sizeof(S1) : 4 bytes, 비트필드의 멤버를 int로 선언했기 때문에

struct S2{
    int b1 : 10;
    int    : 0; // 재설정 : 다음 필드가 비트 32에서 시작하도록 강제
    int b2 : 10;
}; // sizeof(S2) : 8 bytes
```

union

- `union`은 서로 다른 데이터 유형을 동일한 메모리 위치에 저장할 수 있는 특수 데이터 유형
- `union`은 최대 데이터 멤버를 보유하는데 필요한 만큼만 규모를 유지
- `union`은 일종의 겹치는 스토리지

```
union MyUnion{  
    char a;  
    short b;  
    int c;  
};
```



- 구조체와 달리 C++은 anonymous union을 허용함
- C++17은 type-safe union을 표현하기 위해 `std::variant`를 도입함