Structure (P2)

"Notes"

1. func (takes & return) struct.

2. member ⇒ Ptr to this struct.

```
struct bus
{
    int z;
    struct bus * ptr;
};
```

✓

```
struct bus
{
    int z;
    s_bus * ptr;
}s_bus;
```

✗

3.

```
struct Car
{
    int x;
    struct Car y;
};
```

✗ Syntax error

if → another struct ⇒ Valid

↳ Nested struct.

=> Nested structure defin :-

→ Access
↳ (.)

→ init
{ { } }

→ readability :-

ex

→ without Nesting.

```
struct S_employee
{
    char  m_name [50] ;
    int   m_BD_Day;
    int   m_BD_month;
    int   m_BD_Year;
    int   Salary ;
};
```

" الترتيب مهم "

→ with Nesting

```
struct S_Date
{
    int  m_BD_Day;
    int  m_BD_month;
    int  m_BD_Year;
};
struct S_employee
{
    char  m_name [50] ;

    struct S_Date  BD ;

    int  Salary ;
}
```

=> typedef with struct :-

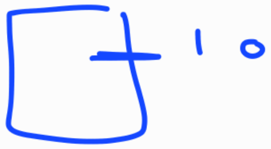
ex

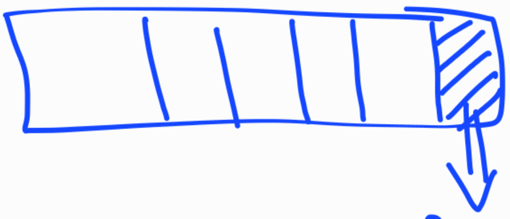
```
struct   my_struct
{
    char  ID;
    int   Sal ;
};
typedef  struct my_struct   emploe ;
```

→ employee X;

```
typedef stru. my_
{
   ══
}employee ;
```

⇒ Bit field





Read bit

GET_BIT(r, r)

① if (((Reg & (1 << $\boxed{7}$)) >> 7)) → 1 / → 0

→ May cause a logical error

② bit field

if (X.Bit7 == 1)

General form → type name : length ;

eg

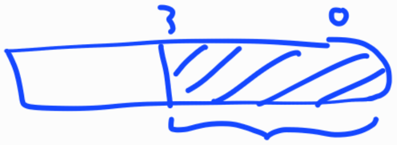
```
struct  Status_type
 {
    unsigned  char   Bit_0 : 1 ;
       ~        ~    Bit_1 : 1 ;
                :
                :
       ~        ~    Bit_7 : 1 ;
 } Status ;
```

Code

```
Status = get_Port ();
if ( Status.Bit_0 )
  {
    ________
    ________
  }
```

Rules

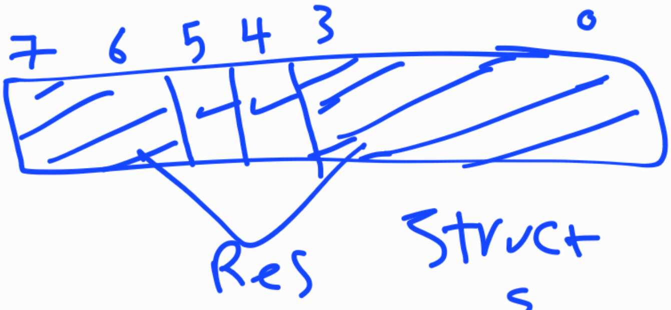
- Access → 1 bit
  Stored → 1 byte
- bit → X address
  X Pointer



Reserved bits

unsigned char : 4 ;

unsigned char Res : 4 ;



```
struct struct_type
{
  unsigned char Res : 4;
  ~      ~   bit_5 : 1 ;
  ~      ~   bit_6: 1 ;
} Status;
```

(Status.bit4 == 1)

✓

⇒ "tricky Codes"

[1]

```
struct emp
{
  float pay;
  unsigned lay_off : 1;
     ~     hourly : 1;
     ~     deduction : 3;
}
```

size = ?

unsigned → int

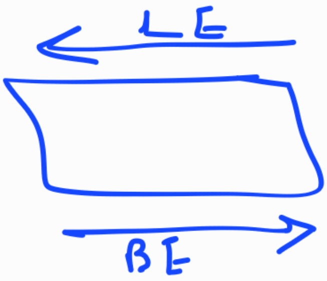
size = 4 + 4 = 8 bytes

hint

- → bit fields Can't be arrayed [i] XX
- → ~ ~ depend on machine

machine → Little En

machine → Big End



```
struct byte
{
  int one : 1;
};
struct byte Var = { 1 };
```

1

one bit

⇒ printf "%d" Var.one

⇓

-1

Ex — → ✓

#bits

1 → $\boxed{1} = \boxed{-1}$

4 → 2

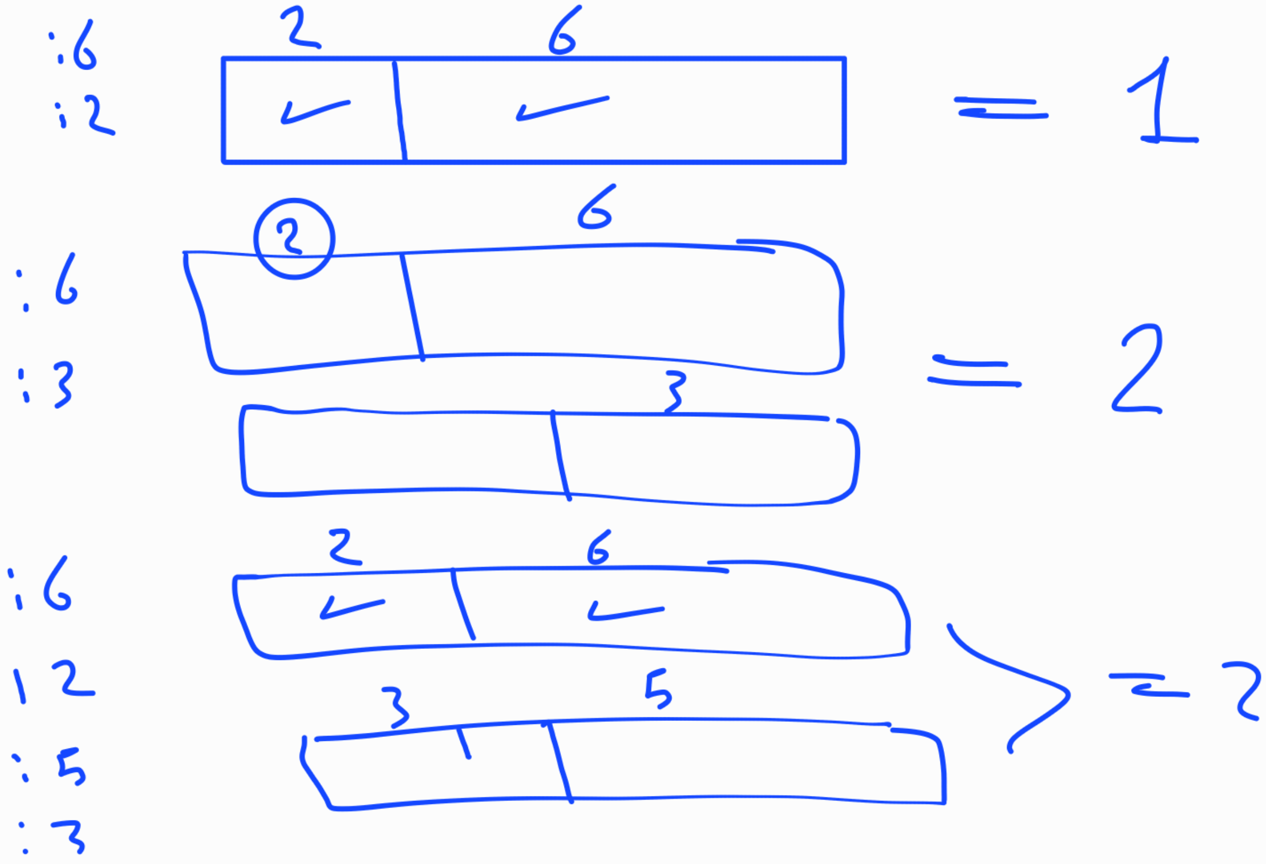
0010 = $\boxed{+2}$

4 → 13 → 2's

1101 → 0011

= $\boxed{-3}$

1 byte = 8 bit



:6
:2
2
6
✓
✓
= 1
②
6
:6
:3
3
= 2
2
6
:6
12
:5
:3
✓
✓
3
5
= 2