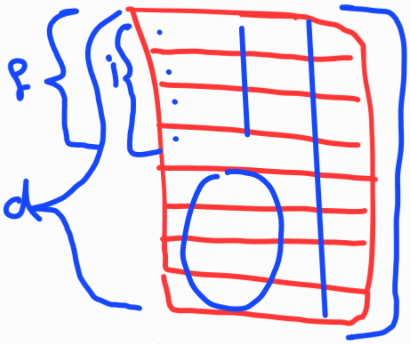
Union

→ User defined data type.

⇒ Support differen overlapped data types.

Union ⇒ members → int, float, double



└→ size

└→ the biggest data type ↗ 8 bytes

Syntax

```
union    UNUM_Value
     {
         int      int_val ;
         float    float_val ;
         double   double_val ;
     };
```

} define new datatype

⇓

No Space @ mem

```
union    UNUM_Value    My_union;
```

ACCess ⟶ dot operator (.)

```
Myunion . float_Val =  ___  ;
```

⇒ shared mem ⇒ overwrite

EX-

→ 3 func ⇒ func

int ← int + int

~ ← double + ~

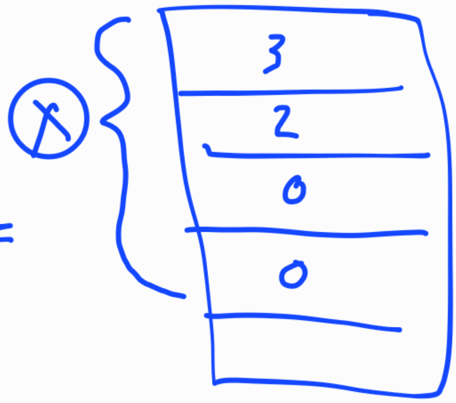
ch ← ch + ch

⇒ using union

⇒ Tricky Codes

→ EX-

T. Arr[0] T. Arr[1] T. X



T. Arr[0] ↓ 3 ↓ ...- 0 0 0 1 1

T. Arr[1] ↓ 2 ↓ ...- 0 0 1 0

$$\ldots - 0\ 0\ 0\ \underbrace{1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1}$$

→ $= 515)_{10}$

⇒ Typedef → Embedded C

Preprocesser Directives

- #include
- #define
- Conditional directive
- #error
#warning
- stringific — #
& Concatin — ##

(.c) → Preprocessor → (.i)

Preprocessor → text replacement => #

#Pragma → Compiler directive

1 #include

→ to include header file (.h)

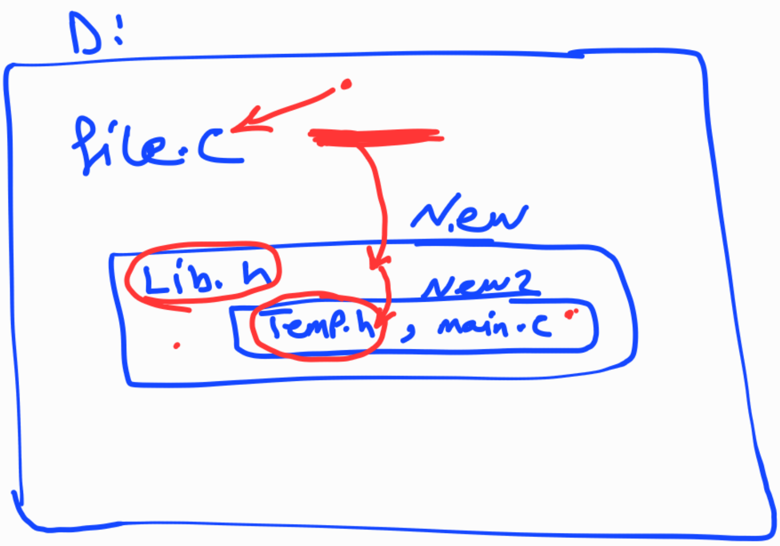
- #include <file.h>
  - Standard Lib (built in Lib)
  - Prepro الـ هيدور فين
- #include "file.h" → Path
  - - Current direc
  - then
  - - Standard Lib

"Path"

- Absolute Path → PC
  - Rule don't use
- Relative Path
  - Current dire

⟹ Relative

EX



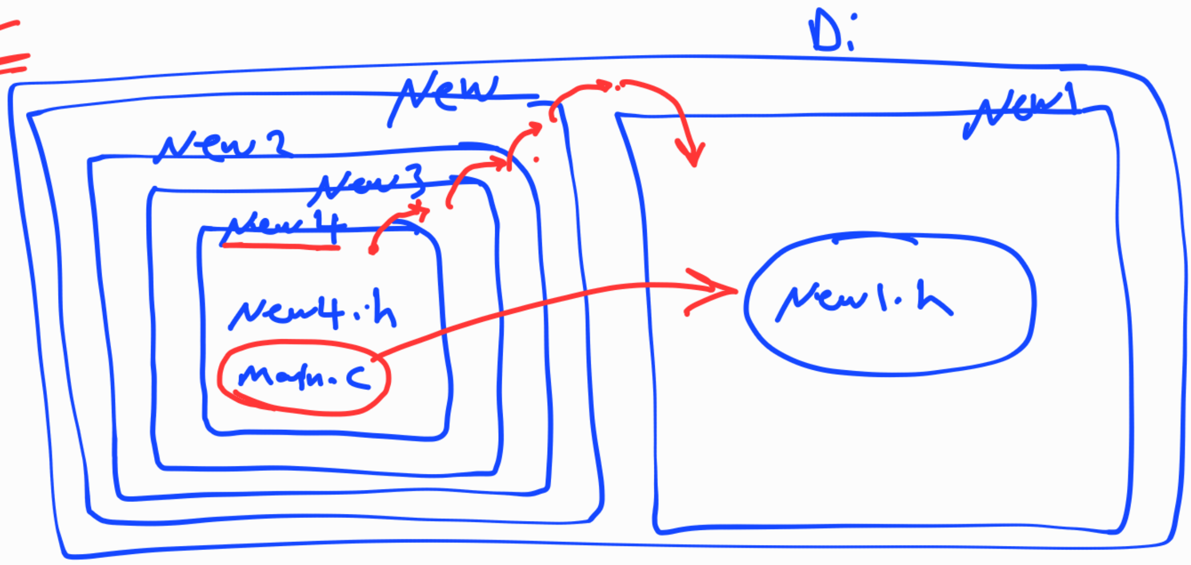
Ex-1 @ file.c ⟶ temp.h

#include "New/New2/Temp.h"

Ex-2 @ main.c ⟶ Lib.h

#include " ../Lib.h"

EX



#include "../../../../New1/New1.h"