

23 September 2022

000 0011 1010 111000 → 0xAF1  
1 3 A F 1

chr → 1 byte

short → 2 byte

integer → dependent upon architecture

long int → 4 bytes

float → 4 bytes

double → 8 bytes

MSB

n-1 bits for signed

0+ve

1-ve

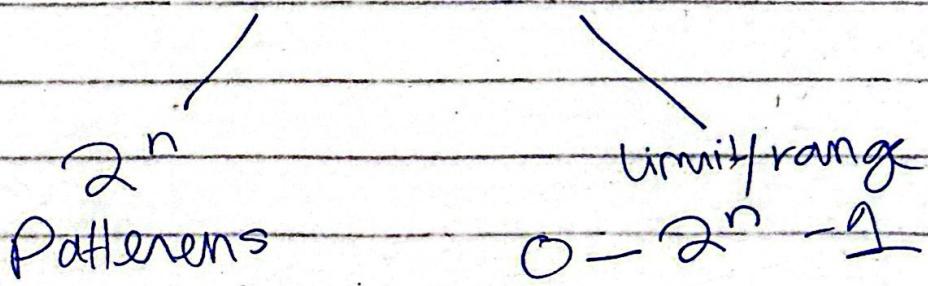
unsigned int :  $2^{31} - 2^0$

limit :  $2^{32} - 1$  unsigned long int

For signed

$$\frac{2^{32}}{2} = 2^{31}$$

N-bit number



if

$$-2^{N-1} = 0 - 2^{N-1} \text{ then } 2^{N+1}$$

→ so have to reduce one number  
from positive side bcs zero  
takes place of one number and  
there is no such thing as  
-ve zero so

$$-2^{31} - 2^{30}$$

for signed short

$$-2^{15} - 2^{14}$$

for unsigned

$$0 - \boxed{2^{16} - 1}$$

for char

signed:  $-2^7 - 2^6$

unsigned  $0 - \boxed{2^8 - 1}$

## Type Casting

```
float x = 3.2;
```

```
int y;
```

$$y = ((int) x) + 2$$

## Functions

output data type      | input data types

```
int main (void)
```

```
{
```

```
} return _____
```

→ indentation doesn't matter in C  
the curly brackets tell us

## Shifting

011011 >> 000110  
011011 << 101100

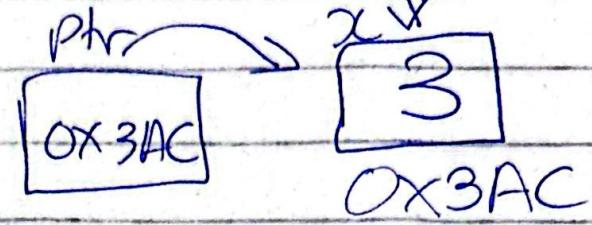
## Pointer

- Stores address
- Points to data

int \*ptr ~~unsigned int~~  
datatype

- The data type of pointer is unsigned integer.

~~X~~ ~~access~~  
int x = 3;



ptr = &x;

to access this value

\*ptr = 3 changes value

ptr = 0X3AC changes address

\*ptr++ => 4

ptr++ => 3BO  
S

4 bytes added to address

Turn 8th bit of led

if want to do through  
int \*bit\_neo(\*int led) { pointers

led = \*led | 0x00000020;

return led

}

int main (void)

int x;

x = 1567.

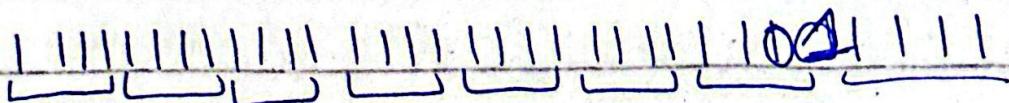
bit set(&x);  
    ^ can skip

}

```
int bit_ckey(led) {
```

```
    led = led & 0xFFFFFBF;  
    return led;  
}
```

~ of previous  
number



0XF F F F F F 0 F

Bit S = 0x00000020

OFF → led &  
~BITS

ON → led |

```
int bit_set(int led, int bit_no){  
    led = led | (0x00000001 << bit_no)  
    return led;  
}
```

$$\text{led} = \text{led} \ \& (\sim(0x00000001 << \text{bit\_no}))$$

constant and volatile

Always  
known  
by  
compiler

data type and type specifier

# add in sum (A+B) ← function like  
# define HELLO S obj of the macro

macros

function like

MACROS  
Object - like

MACROS

0x460A4400

Registers Address

writing into

(Register)

variable

const

Hope you like

#include < .h > } globally recognized header files

#include " .h " } software dependent or self written

#define GPIO\_DIR {  
  0x40004000  
  register  
  S address  
  but must be address

#define GPIO\_DIR\* {  
  dereference  
  (volatile unsigned int \*)  
  0x40004000) }



