

# Computer Programming 1 Lab

2022-12-15

# Outline

- bitwise operations
- Examples
- Exercise11

# bitwise operations

# bitwise operations

## 0x (zero X) prefix

- A prefix to indicate the number is in hexadecimal
- Often used to show memory address.

## example:

```
int a = 114514;
int b = 0x1BF52;    // 114514 in hex
if(a == b){
    printf("True\n");
}
else{
    printf("False\n");
}
return 0;
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c
darkknife@1111cp1:~$ ./a.out
True
darkknife@1111cp1:~$
```

# bitwise operations

## operators

- AND( `&` )
- OR( `|` )
- NOT( `~` )
- XOR( `^` )
- shift operators( `<<` , `>>` )

# bitwise operations

## operator AND( `&` )

- Only when both bits are 1 will result in 1.

	0	1
0	0	0
1	0	1

## example:

```
int a = 0xF;           // a = 0x0000000F
a &= 0xA;              // a = a & 0x0000000A
printf("%d\n", a);
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c
darkknife@1111cp1:~$ ./a.out
10
darkknife@1111cp1:~$
```



F	1	1	1	1
A	1	0	1	0
results	1	0	1	0

## Real world example - IP mask

- Use mask to determine how many devices are there in a subnet.
- Often use 255.255.255.0 (0xFFFFFFFF00), which means there are at most 256 devices on this subnet.

註冊的IP :	140.119.202.●
子網路遮罩Mask :	255.255.255.0
預設閘道Gateway :	140.119.202.254

# bitwise operations

## operator OR( `|` )

- Either one of two bits is 1 will result in 1.

	0	1
0	0	1
1	1	1

## example:

```
int a = 0x6;  
a |= 0xA;    // a = a | 0xA;  
printf("%d\n", a);
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c  
darkknife@1111cp1:~$ ./a.out  
14  
darkknife@1111cp1:~$
```

6	0	1	1	0
A	1	0	1	0
results	1	1	1	0

# bitwise operations

## operator NOT( ~ )

- Not operation will negate the bit

	0	1
~	1	0

## example:

```
int a = 0xFFFFFFFFFA;  
a = ~a;  
printf("%d\n", a);
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c  
darkknife@1111cp1:~$ ./a.out  
5  
darkknife@1111cp1:~$
```

F	1	1	1	1
results	0	0	0	0

A	1	0	1	0
results	0	1	0	1



# bitwise operations

## operator XOR( $\wedge$ )

- Only when one of the bits is 1 will result in 1.

	0	1
0	0	1
1	1	0

## example:

```
int a = 0x00000006;  
a ^= 0x0000000A;    // a = a ^ 0x0000000A;  
printf("%d\n", a);
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c  
darkknife@1111cp1:~$ ./a.out  
12  
darkknife@1111cp1:~$
```

6	0	1	1	0
A	1	0	1	0
results	1	1	0	0

# bitwise operations

operator shift( <<, >> )

- the least-significant bit is lost
- 0 is inserted on the other end

## example:

```
int a = 0x000000F1;
a >>= 2;    // a = a >> 2;
printf("%d\n", a);
a <<= 2;    // a = a << 2;
printf("%d\n", a);
```

## results:

```
darkknife@1111cp1:~$ gcc ./main.c
darkknife@1111cp1:~$ ./a.out
60
240
darkknife@1111cp1:~$
```

0xF1	1	1	1	1	0	0	0	1
$\gg=2$	0	0	1	1	1	1	0	0

0x3C	0	0	1	1	1	1	0	0
$\ll=2$	1	1	1	1	0	0	0	0

# bitwise operations

## example:

Two numbers are given, please generate a bit mask which have 1s in the range between two numbers, and others fill with 0. The bit index count from the lowest bit.

for example:

1. given 2 and 5 for a 8 bit number

-> fill the 2nd, 3rd, 4th bit count from the lowest. -> generate 0001 1100

# bitwise operations

example:

method 1

Run with a for loop, if the number is in between the range, toggle on the bit.

```
for(int i = 0; i < max; i++){  
    if(i >= num1 && i < num2){  
        // .....  
    }  
}
```

Time complexity:  $O(N)$



# bitwise operations

## example:

### method 2

1. make a bit mask of 1s with correct length
2. shift it to the right place.

```
// hi = 5, lo = 2;
mask = 1;           // mask = 0000 0001
mask <=<= (hi-lo);    // mask = 0000 1000
mask -= 1;          // mask = 0000 0111
mask <=<= lo;         // mask = 0001 1100

mask = ((1 <=<= (hi-lo))-1) << lo;
```

Time complexity:  $O(1)$

# Exercise11

# Any Question?

Course? Assignment? Exercise? TA?