

6/19/2025 Comp Prog

(1) Combinations via bitwise operators

(2) bitwise ops

(3) basic example

(4) double plus good

0 1 2 3
{ "mtms", "snickers", "skittles", "sourpatchkids",
4
"hersheys" }

{ "mtms" }

{ "skittles", "hersheys" }

{ }

{ "mtms", "snickers", "skittles", "sourpatchkids", "hersheys" }

	4	3	2	1	0	
	0	1	2	3		
→	0	0	0	0	1	binary number
→	1	0	1	0	0	
→	1	1	1	1	1	

int x 22

32 bits

00000... 10110

2⁴ 2³ 2² 2¹ 2⁰

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Bitwise Operators

a b (and)

10 8 12 $\rightarrow$ 8

$$a = 1010$$
$$\begin{array}{r} 1100 \\ \hline 1000 \end{array}$$
$$a \mid b \quad (\text{or})$$
$$a = 1010$$
$$\begin{array}{r} b = 1100 \text{ } \$1 \\ \hline 1110 \end{array}$$
$$a \wedge b \quad (\text{xor})$$

exclusive or

$$a_1 = 1010$$
$$b = 1100 \quad \wedge$$

0110

$a \gg t \rightarrow$ Right Shift

$$a = \overline{101101100}$$
$$k = 4$$
$$a_{2k} = 10110$$

dividing by 2^k .

$X = 10100101$
 $(1 < 4) = \underline{0010} \underline{0000}$
 off off

$$\nexists a \in K \text{ (left shift)}$$
$$a = 1010 \quad r = 3$$
$$a \ll k = \frac{1010000}{\leftarrow} \quad (\text{multiplying by } 2^k)$$

Goal ~~is~~ in int x find bit k
(from right starting @ 0)

$k=5$ if $(x \& (1 \ll \underline{k}) != 0)$
 else // bit on
 // bit off

```
int bitson (int x) {
```

```
    int res = 0;
    for (int i = 0; i < 32; i++) {
        if ((x & (1 << i)) != 0)
            res++;
    }
    return res;
}
```

```
int res = 0;
while (x > 0) {
    if ((x & 1) != 0)
        res++;
    x = (x >> 1);
}
return res;
```

mask = 19:

4	3	2	1	0
1	0	0	1	1

 mask & (1 << i)

i	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	0	1	2	3	4	1 << i	00001	00001
0	1	2	3	4					
		1 << 1	00010	00010					
		1 << 2	00100	00000					
		1 << 3	01000	00000					
		1 << 4	10000	10000					

10101 + ①
 10101 + ①
 1010 + ⑥
 101 + ①
 10 + ⑥
 1 + ①

cout << CANDY[i]

Double Plus Good Evaluate

3	4	29	8
---	---	----	---

 ↓

0 1 0
+ 0 +

buffer

3

429
8

res 3
+ 429
432
+ 8
440