1. How to store a data.

2. Number system → conversion
→ Data types → Nibble, byte, Short, int, Long
→ Range

3. Bit operators
→ OR,
→ AND
→ XOR
→ left shift
→ Right shift
→ tripple right shift
→ 1's complement
→ 2's complement

4. ON bit
OFF bit
toggle bit
Check if bit is ON or OFF

2	57	
2	28	1
2	14	0
2	7	0
2	3	1
2	1	1
	0	

1. Basic of bits

n = 57

Indexing of bit ←

	5	4	3	2	1	0
binary n =	1	1	1	0	0	1

i = j = k = m = 3

for particular question.

1 1 0 0 0 1

2^5 2^4 2^0

$32 + 16 + 1$ = 49

57
49

seq - 57 → 49

true

Every operation is performed on 'n'

2. Print the number produced on setting its i-th bit.
3. Print the number produced on unsetting its j-th bit.
4. Print the number produced on toggling its k-th bit.
5. Also, Check if its m-th bit is on or off. Print 'true' if it is on, otherwise print 'false'.

2. Right set bit mask →

$n = 58 = (111010)_2$

2	58	
2	29	0
2	14	1
2	7	0
2	3	1
2	1	1
	0	1

Right most

Find that mask

10

right most 1

n = 1 0 1 1 0 1 0 1 0 0

0 0 0 0 0 0 0 1 0 0

rsbm

↳ It is very useful

try to solve it in O(1)

Brute force → ① Start from 0th bit, and check if ith bit is on then it is Right most in number.

② Except that off every bit.

O(1) → ?

$$\underbrace{mask = x \,\&\, x''}_{solution}$$

Math.

x = 1 0 1 1 0 1 0 1 0 0 — right — invert — same

nx = 0 1 0 0 1 0 1 0 1 1

$x'' = x' + 1 = $ 0 1 0 0 1 0 1 1 0 0

inverted — same as x.

0 1 0 0 1 0 1 1 0 0

x = 0 0 0 0 0 0 0 1 0 0

$bm = x \,\&\, -x$ Number ↳ −x

Reverse. right most 1 st

x 0 1 1

& −x Add 1

1 0 0

inverted

0 0 0 1 0 0

$x = 7$

2's compliment of $x = \underbrace{-7}$

Ex:

$n = 7$

binary $\rightarrow$ 1 1 1

2's compliment = 1's compliment + 1

= 1 1 1 1 .... 0 0 0

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ 1

msb = 1

2's compliment = $m =$ 1 1 --- 1 1 0 0 1

$\rightarrow$ -7 $\rightarrow$ Binary form.

2's of m = 1's comp + 1

= 0 0 0 . 0 1 1 0

$\quad\quad\quad\quad\quad$ 1

– 0 0 0 0 1 1 1

= -7

Kernighan's Algorithm :→ Find no. of ON bits in a number 'n'

no. of ON bit = ?

n = 1 1 0 1 0 0 1 0 0 1 1 0 0 1 1 0 0

Bruteforce

1. Iterate on every bit
2. Check if bit is on then increment count
3. Print count

$\boxed{\text{Complexity} \to 32 \text{ Iteration.}}$

1. can we find right most one bit in O(1)
2. Can we OFF Right most one bit in O(1)

rsbm = $(x \,\&\, -x)$ $\equiv$ $x \,\&\, x''$ ← 2's comp.

$\boxed{\text{Complexity} \quad \text{no. of 1's Iteration}}$

A − B

A + (−B) = A − B

1 0

rsbm =

right most set bit mask

$x = x - \text{rsbm}$ } code.

cnt

last one zero

Repeat until n is not equal to '0'

n = 1 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0

rsbm = bit Print → 2 to set O(1) $\nleftarrow$ rsbm = 1 0

cnt

21 20

n − rsbm = 1 1 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0

Kernighan's Algo

$x - \underbrace{rsbm}$

$\underbrace{x \quad + \quad 2\text{'s compliment of } rsbm}$

$x + (-rsbm)$

$x = \; 1 \; 1 \; 0 \; 0 \; \cancel{1} \; 0 \; 0 \; 0$

$x - rsbm \quad 0 \; 0 \; 0 \; 0 \; \cancel{1} \; 0 \; 0 \; 0$

remove rightmost 1 $\rightarrow$ $1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0$

$x = \ldots 0 \; 0 \; 1 \; 1 \; 0 \; 0 \; 1 \; 0 \; 0 \; 0$

$(-rsbm) = 1 \ldots 1 \; 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; 0$

$0 \ldots 0 \; 0 \; 1 \; 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0$

$rsbm = \underbrace{1 \; 0 \; 0 \; 0} \longrightarrow 2\text{'s compl} = -rsbm$

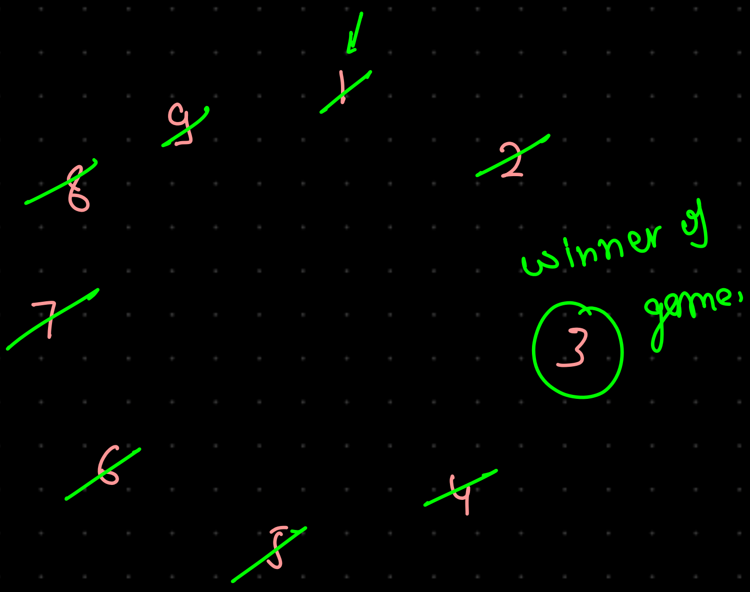
$= 1\text{'s comp} + 1$

$= 1 \; 1 \; 1 \ldots 0 \; 1 \; 1 \; 1 + \quad 1$

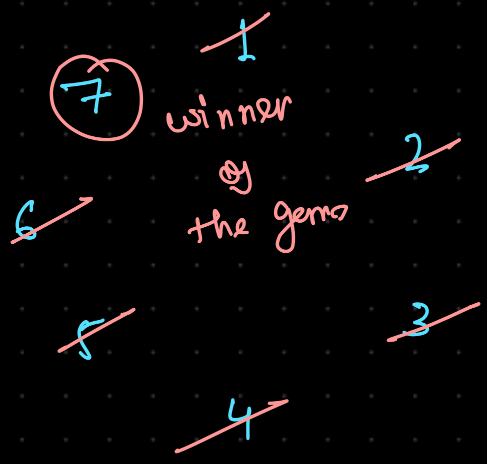
2's comp of rsbm $= 1 \ldots 1 \; 1 \; 1 \; 1 \; 0 \; 0 \; 0$

Josephus Special :→

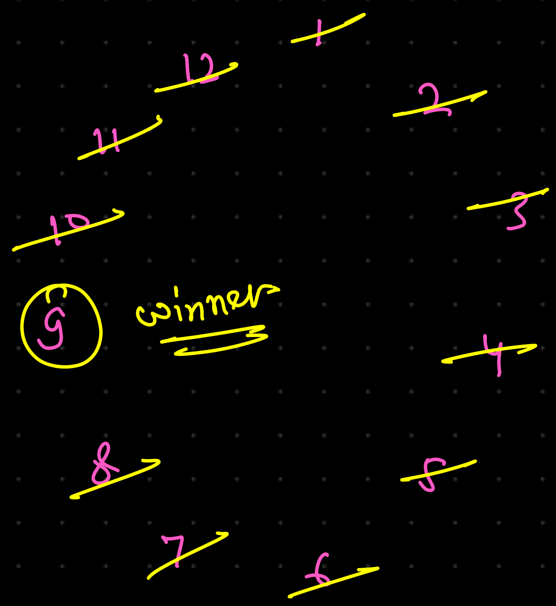
n = 9



n = 7



n = 12



Remaining player → ?

given = n → no. of plays in game.

$\underbrace{n = 2^{x} + l}$ x : max. power of 2 in n.

Remaining player = $\underbrace{2l + 1}$

1. What
2. How
3. Why.

n = 12

1 2 3 4 5 6 7 8 9 10 11 12

9 r. player

$n = 2^x + l$

$= 2^3 + 4 \rightarrow l$

$= 8 + 4 = 12$

Remaining player $= 2l + 1$

$= 4 \times 2 + 1$

$= 8 + 1$

$= 9$

n = 17

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

3 r.p'

$17 = 2^x + l = 2^4 + 1$ l

r.p $= 2l + 1$

$= 2 \times 1 + 1 = 3$

n = 15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

15 r.p

$15 = 2^x + 1$

$15 = 2^3 + 7$ l

R. Player $= 2l + 1$

$= 2 \times 7 + 1$

$= 15$

n=12

$12 = 2^3 + 4$. $l=4$, r.p = $4 \times 2 + 1$ = 9

Binary Form.

1 → 0 0 0 1

2 → 0 0 1 0

3 → 0 0 1 1

4 → 0 1 0 0

5 → 0 1 0 1

6 → 0 1 1 0

7 → 0 1 1 1

8 → 1 0 0 0

9 → 1 0 0 1

10 → 1 0 1 0

11 → 1 0 1 1

12 → 1 1 0 0

remain.

sur

0 0 0 1 =0

0 0 1 1

0 1 0 1

0 1 1 1

1 0 0 1

1 0 1 1

Next surviving bit

1st position 0 ⇒ 2nd posi

0 → surviving bit

$2^x + l$ res = 1 0 0 1

① Kill all number having last bit is 0

survy = 0

↓

0 0 0 1

0 1 0 1

1 0 0 1

1 is surviv's bit

↓

0 0 0 1

$\boxed{1 0 0 1}$

9

$n = \underbrace{2^x + l}$

where x is max. power of 2 in number 'n'.

OR

↳ x is left most ON bit in a number. 'n'.

n = 9

1 → 0 0 0 1
2 → 0 0 1 0
3 → 0 0 1 1
4 → 0 1 0 0
5 → 0 1 0 1
6 → 0 1 1 0
7 → 0 1 1 1
8 → 1 0 0 0
9 → 1 0 0 1

last num

res = 0 0 0 1 1

remaining player

surviving bit = 1

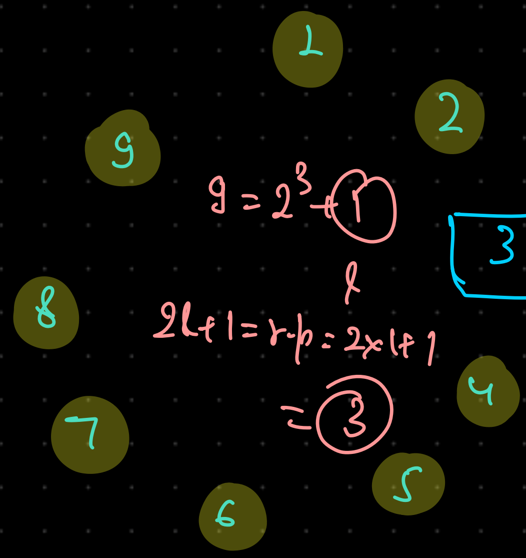
0 0 0 1
0 0 1 1
0 1 0 1
0 1 1 1
1 0 0 1

surviving bit = 0

0 0 1 1 = 3

0 1 1 1

NOTE: last number will help you to know next surviving bit



$9 = 2^3 + 1$

l

$2l+1 = r.p = 2 \times 1 + 1$

$= 3$

$\boxed{3}$ Remaining play.

n = 17

1 → 0 0 0 0 1
2 → 0 0 0 1 0
r.p. $\boxed{3}$ → 0 0 0 1 1
4 → 0 0 1 0 0
5 → 0 0 1 0 1
6 → 0 0 1 1 0
7 → 0 0 1 1 1
8 → 0 1 0 0 0
9 → 0 1 0 0 1
10 → 0 1 0 1 0
11 → 0 1 0 1 1
12 → 0 1 1 0 0
13 → 0 1 1 0 1
14 → 0 1 1 1 0
15 → 0 1 1 1 1
16 → 1 0 0 0 0
17 → 1 0 0 0 1

res = 0 0 1 1

Ans

rem. p. — 1 is surviving bit

0 0 0 0 1
0 0 0 1 1
0 0 1 0 1
0 0 1 1 1
0 1 0 0 1
0 1 0 1 1
0 1 1 0 1
0 1 1 1 1
1 0 0 0 1

Rem. play. — surviving bit

0 0 0 1 1
0 0 1 1 1
0 1 0 1 1
0 1 1 1 1

surviving bit = 0

$\boxed{0\ 0\ 0\ 1\ 1}$
0 1 0 1 1

Remaining player = 0 0 0 1 1

3

$2^x + l = n.$

$\underbrace{2^x}$

$\hookrightarrow A + l = n.$

$\underbrace{l = n - A}$

$\text{result} = \underbrace{2l + 1}$

$\boxed{n = \underbrace{17}}$

$i = 1$

$i = i * 2 = 2$

$i = i * 2 = 4$

$i = i * 2 = 8$

$i = i * 2 = 16 \qquad 2^x + l = 17$

$i = i * 2 = 32$ greater

$2^x + l = 17$

$l = 17 - 2^x = 17 - 16 = 1$

$\Rightarrow l = 1$

$\text{result} = 2l + 1 = 3$

Gray Code: →

Input n → no. of bits.

gray code which can generate using n-bit.

order ↳ bits arrangement possible after single toggle bit

dependent

problem

n=1 → 0
1

[0, 1]

n=2 → 0 0
0 1
1 1
1 0

[0 0 01 11 10]

n=3

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

000 001

011 010

110 111

101 100

Initial bit position

n = 3.

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0

n=4 →

0 0 0 0
0 0 0 1
0 0 1 1
0 0 1 0
0 1 1 0
0 1 1 1
0 1 0 1
0 1 0 0
1 1 0 0
1 1 0 1
1 1 1 1
1 1 1 0
1 0 1 0
1 0 1 1
1 0 0 1
1 0 0 0

n = 1 → [0 , 1]

n = 2 → [0 0   0 1   1 1   1 0]

0 attach   attach 1 in reverse

n=3 →

0	1	3	2
0 0 0	0 0 1	0 1 1	0 1 0
6	7	5	4
1 1 0	1 1 1	1 0 1	1 0 0

on second bit

n=4 →

0000	0001	0011	0010
0110	0111	0101	0100
1100	1101	1111	1110
1010	1011	1001	1000

Schedule

Old Schedule

Tue → 9-12
thur → 9-12
Sat → 11-2
Sum → 11-2

DSA

9-12
9-12

System Design.

For this week

Thursday → 9-12 ✓
Frid → 9-12
Satu → 11-2 → System Design
Sum → OF → System Design

From next week

New Schedule

Mon → 9-12
Tue → 8-12
Wed → 8-12
Thurs → 9-12
Fri → 9-12
Sat → 9-12
Sun → 9-12

DSA
↳ Shresh

System Design

Sumeet Sir