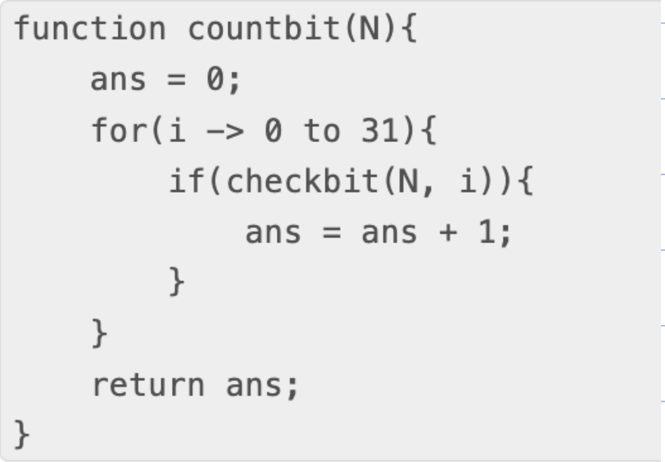
Question: Count # set bits

N = 10 ⇒ 1010 ⇒ 2

N = 8 ⇒ 1000 ⇒ 1

N = 7 ⇒ 0111 ⇒ 3

Approach1: (will not work in Python)



```
function countbit(N){
    ans = 0;
    for(i -> 0 to 31){
        if(checkbit(N, i)){
            ans = ans + 1;
        }
    }
    return ans;
}
```

Approach2:

N = 22

$\underset{4}{1}\ \underset{3}{0}\ \underset{2}{1}\ \underset{1}{1}\ \underset{0}{0}$

↓ >> 1

1 0 1 1

↓ >> 1

1 0 1

↓ >> 1

1 0

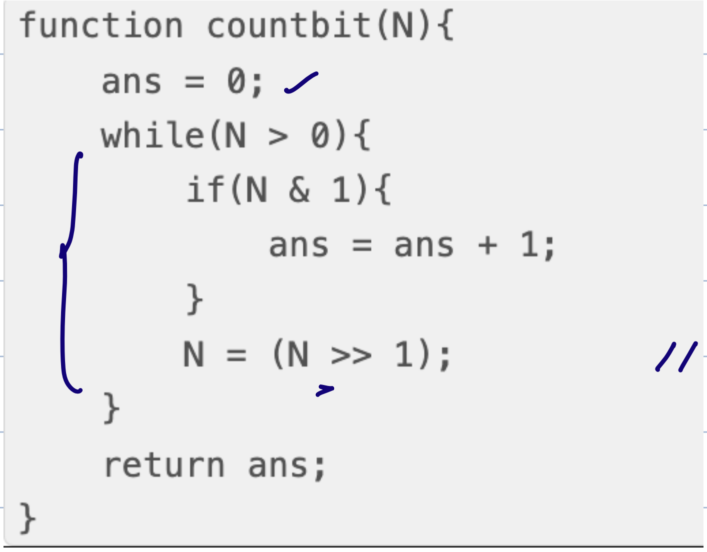
↓ >> 1

1

↓ >> 1

0

ans = 0 + 1 + 1 + 1



```
function countbit(N){
    ans = 0;
    while(N > 0){
        if(N & 1){
            ans = ans + 1;
        }
        N = (N >> 1);
    }
    return ans;
}
```

// N = N/2

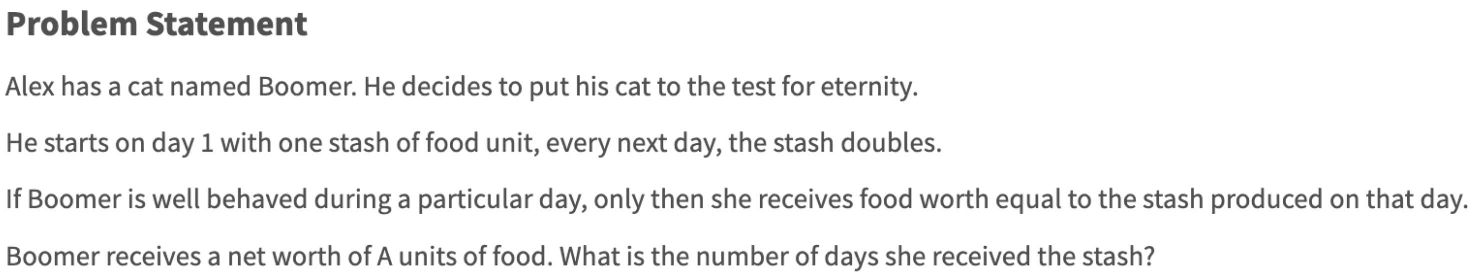
T.C: $O(\log N)$

S.C: $O(1)$

$$N \rightarrow \frac{N}{2} \rightarrow \frac{N}{2^2} \rightarrow \frac{N}{2^3} \rightarrow \cdots\cdots\cdots \underset{k}{1} \rightarrow 0$$

$$\frac{N}{2^K} = 1 \Rightarrow N = 2^K \Rightarrow \boxed{K = \log_2 N}$$

Questn:



Problem Statement

Alex has a cat named Boomer. He decides to put his cat to the test for eternity.

He starts on day 1 with one stash of food unit, every next day, the stash doubles.

If Boomer is well behaved during a particular day, only then she receives food worth equal to the stash produced on that day.

Boomer receives a net worth of A units of food. What is the number of days she received the stash?

Day 1 : 1 ✓

Day 2 : 2 ✓

Day 3 : 2^2

Day 4 : 2^3

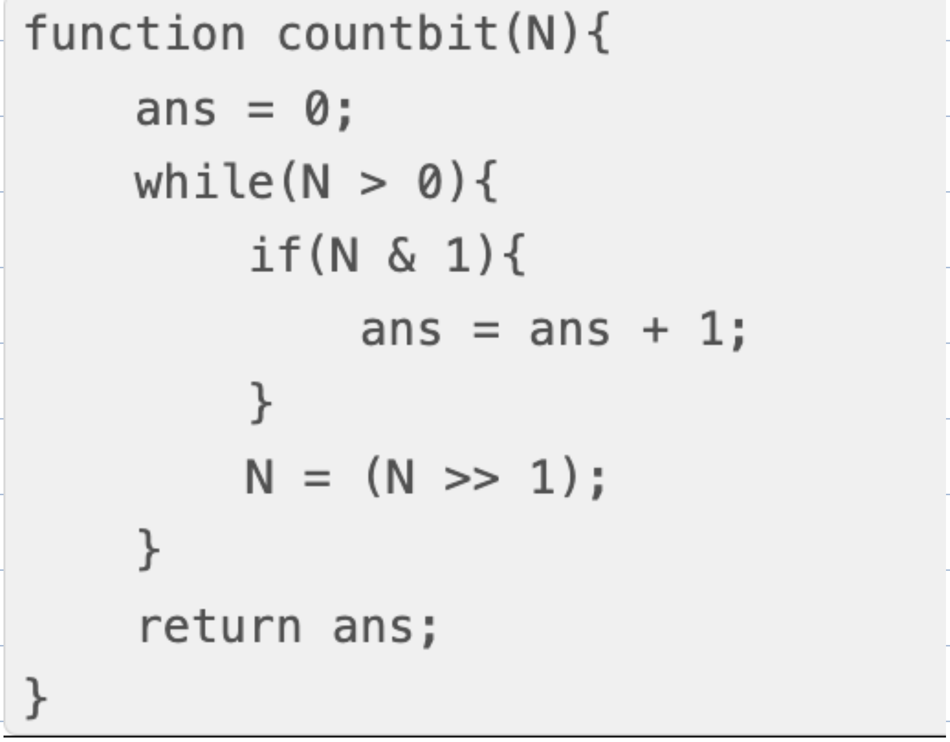
Day 5 : 2^4

Day 6 : 2^5

$\vdots \qquad \vdots$

$$A = 10 \Rightarrow \underset{\text{Day } 4}{\underset{\downarrow}{2^3}} + \underset{\text{Day } 2}{\underset{\downarrow}{2^1}} \Rightarrow \underset{\downarrow}{\underset{3}{1}}\ \underset{\downarrow}{\underset{2}{0}}\ \overset{1}{\underset{\downarrow}{\underset{1}{1}}}\ \underset{\cancel{\downarrow}}{\underset{0}{0}}$$

Count # of set bits



```
function countbit(N){
    ans = 0;
    while(N > 0){
        if(N & 1){
            ans = ans + 1;
        }
        N = (N >> 1);
    }
    return ans;
}
```

Question: All elements occur thrice except one.
Find the unique element

$A[i] \leq 10^9 \Rightarrow$ (32 bits)

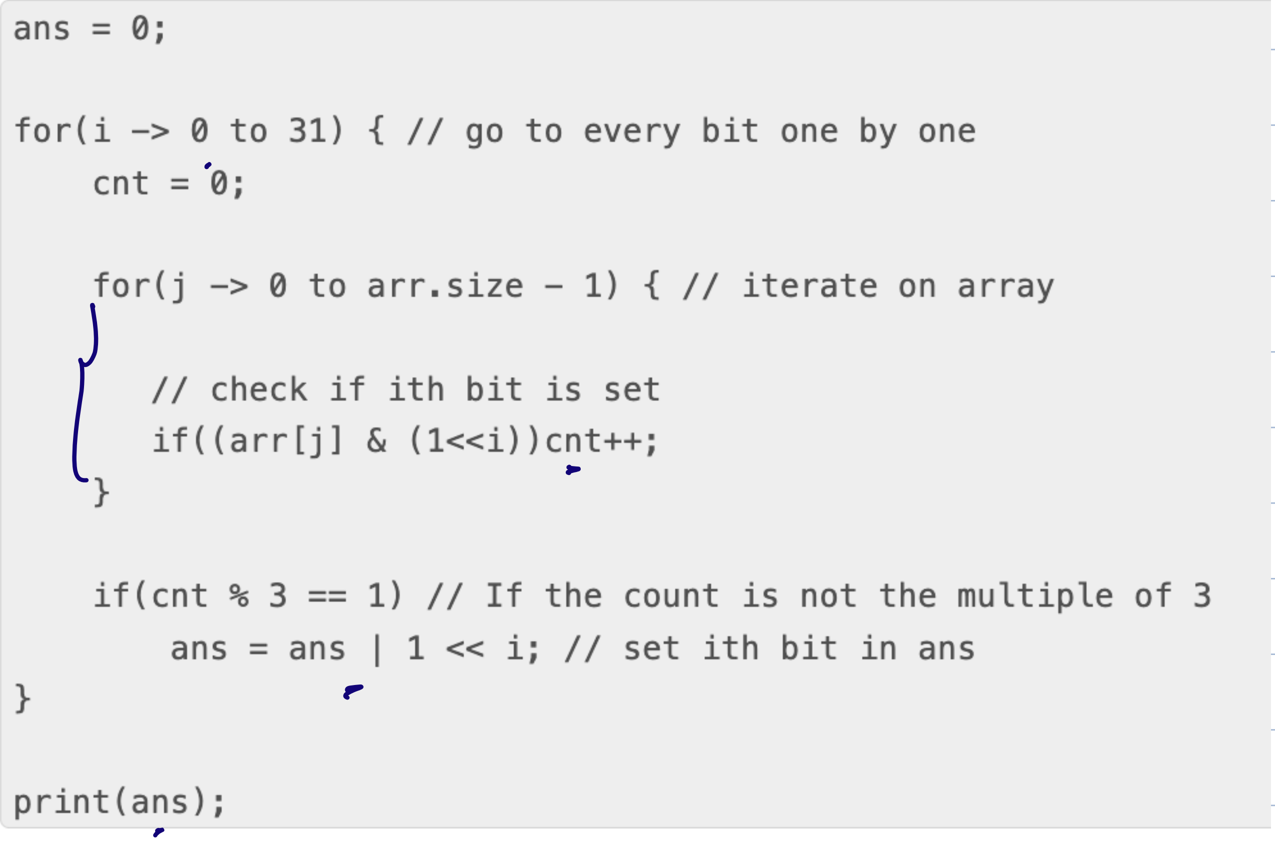
A = [4 5 5 4 1 6 6 4 5 6]

Approach:

A: [3 2 2 3 5 3 2]

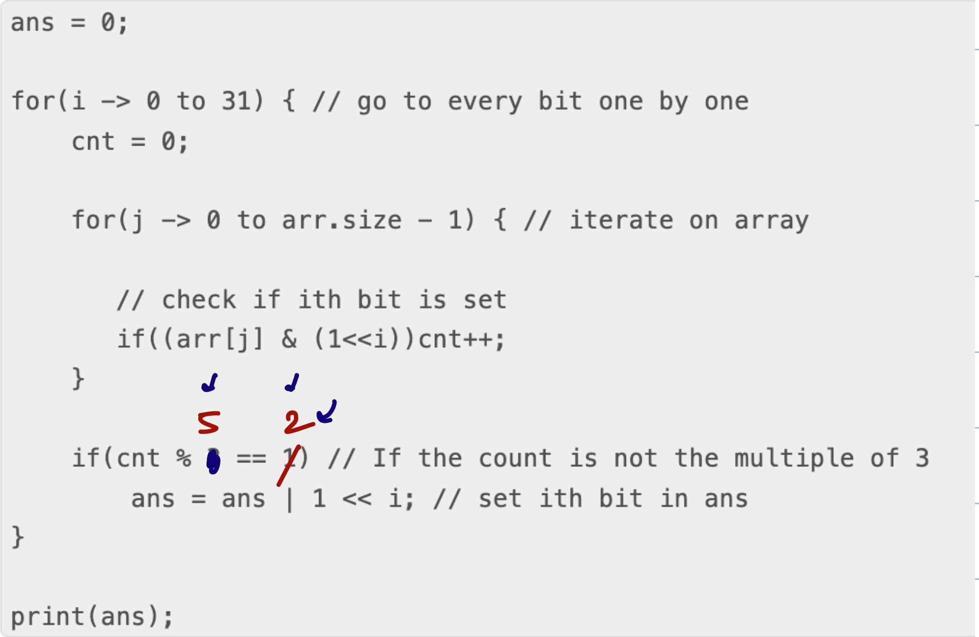
	2	1	0
3 :	0	1	1
2 :	0	1	0
2 :	0	1	0
3 :	0	1	$\textcircled{1}$
5 :	1	0	1
3 :	0	1	1
2 :	0	1	0
	1%3	6%3	4%3
		0	1

ans : $\underset{2}{1}$ $\underset{1}{0}$ $\underset{0}{1}$ => $\boxed{5}$



```
ans = 0;

for(i -> 0 to 31) { // go to every bit one by one
    cnt = 0;

    for(j -> 0 to arr.size - 1) { // iterate on array

       // check if ith bit is set
       if((arr[j] & (1<<i))cnt++;
    }

    if(cnt % 3 == 1) // If the count is not the multiple of 3
        ans = ans | 1 << i; // set ith bit in ans
}

print(ans);
```

Question: All elements occur 5 times except 1 except element which occurs twice



```
ans = 0;

for(i -> 0 to 31) { // go to every bit one by one
    cnt = 0;

    for(j -> 0 to arr.size - 1) { // iterate on array

       // check if ith bit is set
       if((arr[j] & (1<<i))cnt++;
    }

    if(cnt % ~~3~~ 5 == ~~1~~ 2) // If the count is not the multiple of 3
        ans = ans | 1 << i; // set ith bit in ans
}

print(ans);
```

#sets = 5x + $\boxed{2}$

5x

Question:

Problem Statement

You are given an array consisting of 0s and 1s. Your task is to calculate the number of subarrays for which the bitwise OR of all the elements in the subarray is 0.

A = [1 0 0 1]

0 1 2 3

$\frac{4 \times 5}{2} = 10$

[1, 1] [2, 2] [1, 2] ⇒ 3

A = [1 0 0 1 1 0 1 0 0 0 1 0 0]

0 1 2 3 4 5 6 7 8 9 10 11 12

$\frac{2(3)}{2} = 3$ $\frac{1 \times 2}{2} = 1$ $\frac{3 \cdot 4}{2} = 6$ $\frac{2 \cdot 3}{2} = 3$

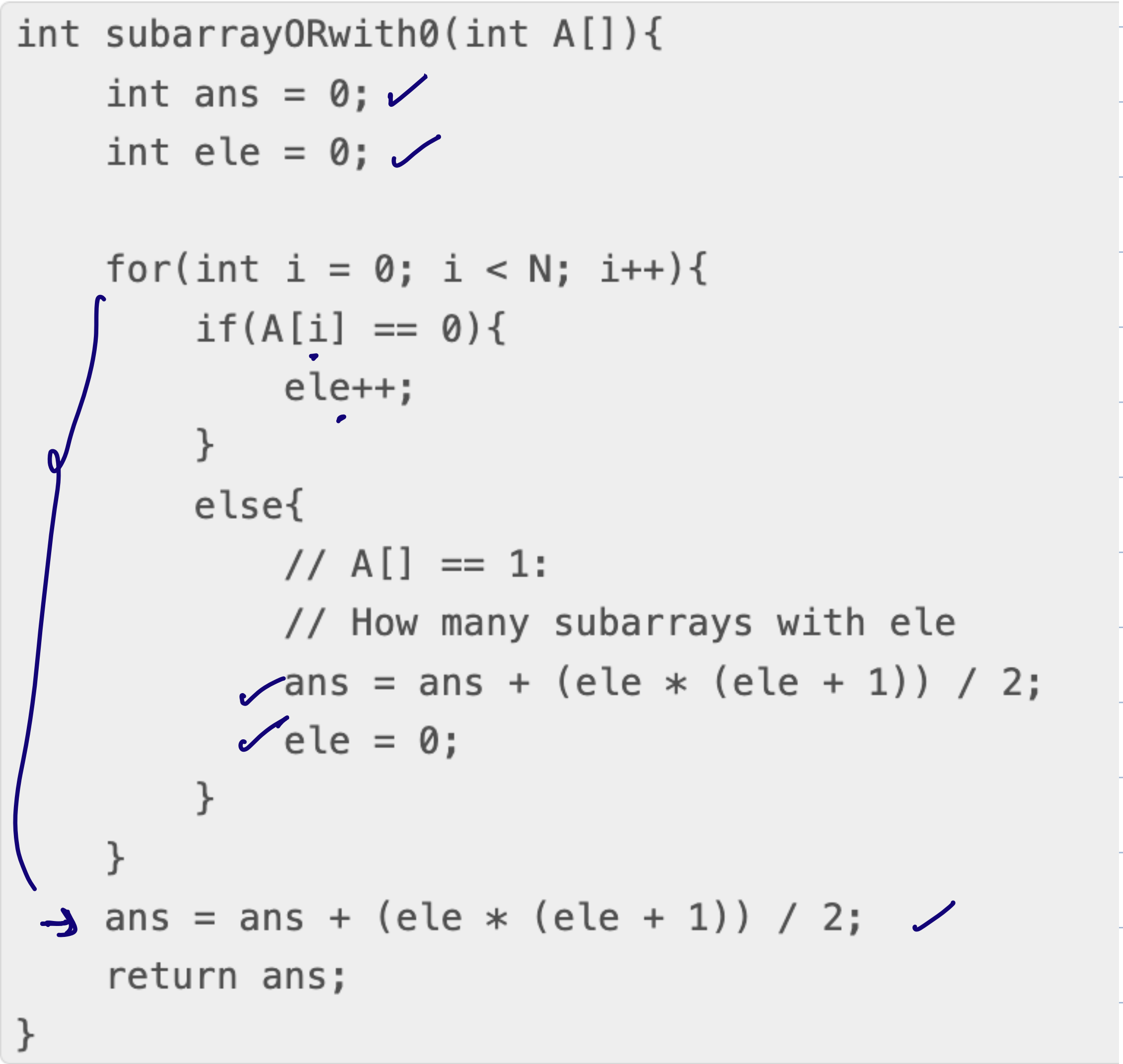
13

A = [1 0 0 1 1 0 1 0 0 0 1 0 0]

count 0 = ~~0~~ ~~1~~ ~~2~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ 2

ans = ~~0~~ 3 + 1 + 6 + 3

```
if( A[i] == 0) count_0++;
else {
  ✓ ans = ans + count_0 * (count_0 + 1) / 2;
  ✓ count_0 = 0;
}
```



```
int subarrayORwith0(int A[]){
    int ans = 0;
    int ele = 0;

    for(int i = 0; i < N; i++){
        if(A[i] == 0){
            ele++;
        }
        else{
            // A[] == 1:
            // How many subarrays with ele
            ans = ans + (ele * (ele + 1)) / 2;
            ele = 0;
        }
    }
    ans = ans + (ele * (ele + 1)) / 2;
    return ans;
}
```

T.C: O(N)

S.C: O(1)

Question: #subarray with BITWISE OR 1

#Total Subarray = {Subarrays with OR 0}

+

{Subarrays with OR 1}