2.1.1. Count Number of Set Bits (1s) In An Integer

08:45

You need to count how many bits are set to 1 in the binary representation of a given integer. Use bitwise operators (& and >>) to achieve this.

Input Format:

- A single integer n

Output Format:

- Print the count of set bits in the integer.

Constraints:

- $0 \leq n \leq 10^6$

Sample Test Cases +

Explorer

CTC758.c

Submit

Debugger

```c
#include<stdio.h>
void main()
{
    int n;
    scanf("%d",&n);
    int count = 0;
    while(n>0){
        count += n & 1;
        n = n >> 1;
    }
    printf("%d\n",count);
}
```

Average time	Maximum time
0.065 s	0.077 s
64.67 ms	77.00 ms

2 out of 2 shown test case(s) passed

1 out of 1 hidden test case(s) passed

Test case 1 77 ms

Debug

Expected output	Actual output
5	5
2	2

Test case 2 58 ms

Terminal Test cases

2.1.2. Check If A Given Bit is Set or Not

13:27

Given an integer and a bit position, determine if that bit is set (1) or not (0). Bit positions are counted from 0 (least significant bit).

Input Format:

- Two space-separated integers n and pos

Output Format:

- Print 'Set' if the bit is 1, otherwise print 'Not Set'.

Constraints:

- $0 \leq pos \leq 31$

Sample Test Cases +

Explorer

CTC759.c

Submit

Debugger

```c
#include<stdio.h>
void main()
{
    int num,pos;
    scanf("%d %d",&num,&pos);
    if(num & (1 << pos)){
        printf("Set");
    }
    else{
        printf("Not Set");
    }
}
```

Average time **0.038 s** 38.33 ms

Maximum time **0.067 s** 67.00 ms

2 out of 2 shown test case(s) passed

1 out of 1 hidden test case(s) passed

Test case 1 37 ms

Debug

Expected output	Actual output
5 0	5 0
Set	Set

Test case 2 67 ms

Terminal Test cases