3.1.1. Toggle the kth Bit Of A Number

03:51

Write a C program that toggles (flips) the k^{th} bit of a given integer. Use the XOR (^) operator to achieve this.

Input Format:

- Two space-separated integers n and k

Output Format:

- Print the new number after toggling the bit.

Constraints:

- $0 \leq k \leq 31$

Sample Test Cases +

Explorer

CTC760.c

Submit

```c
#include<stdio.h>
void main()
{
    int num,pos;
    scanf("%d %d",&num,&pos);
    int result = num ^ (1<<pos);
    printf("%d",result);
}
```

Debugger

Average time	Maximum time
0.052 s	0.057 s
51.67 ms	57.00 ms

2 out of 2 shown test case(s) passed

1 out of 1 hidden test case(s) passed

Test case 1 51 ms — Debug

Expected output	Actual output
5 1	5 1
7	7

Test case 2 47 ms

Terminal Test cases

< Prev Reset Submit Next >

3.1.2. Multiply a Number by 2 power n Using Bitwise Shift

03:40

You are required to multiply a given integer by 2 raised to the power n using bitwise left shift (<<).

Input Format:

- Two space-separated integers x and n

Output Format:

- Print the result after multiplication.

Constraints:

- $0 \leq n \leq 10$

Sample Test Cases +

Explorer

CTC761.c

Submit

```c
#include<stdio.h>
void main()
{
    int num,power;
    scanf("%d %d",&num,&power);
    int result = num << power;
    printf("%d",result);
}
```

Debugger

Average time	Maximum time
0.060 s	0.078 s
59.67 ms	78.00 ms

2 out of 2 shown test case(s) passed

1 out of 1 hidden test case(s) passed

Test case 1 51 ms

Debug

Expected output	Actual output
5 3	5 3
40	40

Test case 2 50 ms

Terminal Test cases

< Prev Reset Submit Next >