**Check whether K-th bit is set or not: -**

**Easy Accuracy: 52.75% Submissions: 134K+ Points: 2**

Given a number **N**and a bit number **K**, check if **Kth** index bit of **N**is set or not. A bit is called set if it is 1. Position of set bit '1' should be indexed starting with 0 from LSB side in binary representation of the number.  
**Note:** Index is starting from 0. You just need to return **true**or **false**, driver code will take care of printing "Yes" and "No".

**Example 1:**

**Input**:   
N = 4  
K = 0

**Output**:   
No

**Explanation**:   
Binary representation of 4 is 100, in which 0th index bit from LSB is not set. So, return false.

**Example 2:**

**Input**:   
N = 4  
K = 2

**Output**:   
Yes

**Explanation**:   
Binary representation of 4 is 100, in which 2nd index bit from LSB is set. So, return true.

**Example 3:**

**Input**:   
N = 500  
K = 3

**Output**:   
No

**Explanation**:   
Binary representation of 500 is 111110100, in which 3rd index bit from LSB is not set. So, return false.

**Your task:**

You don't have to read input or print anything. Your task is to complete the function**checkKthbit**that takes **n**and**k**as parameters and returnseither **true**(if kth bit is set) or **false**(if kth bit is not set).  
 **Expected Time Complexity:** O(1).  
**Expected Auxiliary Space:** O(1).  
  
**Constraints:**  
1 ≤ N ≤ 109  
0 ≤ K ≤ 31

**Code: -**

//{ Driver Code Starts

//Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User function Template for C++

class Solution

{

public:

// Function to check if Kth bit is set or not.

bool checkKthBit(int n, int k)

{

// Your code here

// It can be a one liner logic!! Think of it!!

return n & (1 << k);

}

};

//{ Driver Code Starts.

// Driver Code

int main()

{

int t;

cin>>t;//taking testcases

while(t--)

{

long long n;

cin>>n;//input n

int k;

cin>>k;//bit number k

Solution obj;

if(obj.checkKthBit(n, k))

cout << "Yes" << endl;

else

cout << "No" << endl;

}

return 0;

}

// } Driver Code Ends

**T.C: - O(1)**

**S.C: - O(1)**