**Lucky Numbers: -**

Easy Accuracy: 30.35% Submissions: 72K+ Points: 2

**Lucky numbers** are subset of integers. Rather than going into much theory, let us see the process of arriving at lucky numbers,  
Take the set of integers  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,……  
First, delete every **second**number, we get following reduced set.  
1, 3, 5, 7, 9, 11, 13, 15, 17, 19,…………  
Now, delete every **third** number, we get  
1, 3, 7, 9, 13, 15, 19,….….  
Continue this process indefinitely……  
Any number that does **NOT** get deleted due to above process is called “**lucky**”.

You are given a number **N**, you need to tell whether the number is lucky or not. If the number is lucky return 1 otherwise 0.

**Example 1:**

**Input:**

N = 5

**Output:** 0

**Explanation:** 5 is not a lucky number

as it gets deleted in the second

iteration.

**Example 2:**

**Input:**

N = 19

**Output:** 1

**Explanation:** 19 is a lucky number because   
it does not get deleted throughout the process.

**Your Task:**  
You don't need to read input or print anything. You only need to complete the function **isLucky()**that takes N as parameter and returns either False if the N is not lucky else True.

**Expected Time Complexity:**O(sqrt(N)).  
**Expected Auxiliary Space:**O(sqrt(N)).

**Constraints:**  
1 <= N <= 105

**Code: -**

//{ Driver Code Starts

//Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

//User-function template for C++

// Complete the function

// n: Input n

// Return True if the given number is a lucky number else return False

class Solution{

public:

bool isLucky(int n) {

int divisor = 2, dividend = n;

while(dividend / divisor != 0){

if(dividend % divisor == 0)

return false;

dividend = dividend - (dividend / divisor);

++divisor;

}

return true;

}

};

//{ Driver Code Starts.

signed main(){

int T;

cin>>T;

while(T--){

int n;

cin>>n;

Solution obj;

//calling isLucky() function

if(obj.isLucky(n))

cout<<"1\n";//printing "1" if isLucky() returns true

else

cout<<"0\n";//printing "0" if isLucky() returns false

}

}

// } Driver Code Ends

**T.C: - O(√n)**

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