**Check squares**

Show Topic Tags   

Given an integer N, check if it is possible to represent it as a function(a, b) such that : a2 + b2= N where "a" and "b" are whole numbers.

**Input:**  
The first line of input contains an integer T, denoting number of test cases. Then T lines follow, each of them consisting of exactly one integer N.  
  
**Output:**  
For each test case output 1 if it is possible to represent given number as a sum of two squares and 0 if it is not possible.  
  
**Constraints:**  
1<=T<=100  
1<=N<=106  
  
**Example:  
Input:**  
3  
1  
2  
7  
**Output:**  
1  
1  
0  
  
**Explanation  :**  
1 = 02 + 12   : Hence Possible : Output =  1    
2 = 12 + 12   : Hence Possible : Output =  1

\*\*For More Examples Use Expected Output\*\*

Contributor: Saksham Raj Seth

<http://practice.geeksforgeeks.org/problems/check-squares/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Arrays;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

int ans =0;

for(int i =0; ans == 0 && i<= Math.sqrt(n); i++) {

for(int j =0; j <= Math.sqrt(n); j++) {

if(i\*i + j\*j == n) {

ans = 1;

break;

}

}

}

System.out.println(ans);

}

}

}