**Cube root of a number**

[maths](http://www.practice.geeksforgeeks.org/tag-page.php?tag=maths&isCmp=0)

Given a number **n**, find the cube root of**n**.

**Note:**We need to print the floor value of the result.

**Input:**

The first line of input contains a single integer **T** denoting the number of test cases. Then**T** test cases follow. Each test case consist of one line. The first line of each test case consists of an integer**n**.

**Output:**

Corresponding to each test case, in a new line, print the cube root of **n** ( if the answer is in decimal, print its **floor**value).

**Constraints:**

1 ≤ T ≤ 100  
1 ≤ n ≤ 100000  
  
**Example:**

**Input**  
2  
3  
8

**Output**  
1  
2

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1098>

#include <iostream>

#include <stdio.h>

#include <math.h>

#include <vector>

#define ll long long int

using namespace std;

int main() {

int t;

scanf("%d", &t);

while(t--) {

int n ;

cin >> n;

double result = floor( pow(n, (1.0 / 3.0)) );

cout << result << endl;

}

return 0;

}