**Can you Guess ?**

Attempted by: **847**

/

Accuracy: **60%**

/

Maximum Score: **10**

/

10 Votes

Tag(s):

Very-Easy, Very-Easy

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

No problem statement.

Find the logic from the given sample input/output.

And answer Q queries.

Constraints :

1 <= Value <= 100000

1<=nunber of query<=10000

**SAMPLE INPUT**

8

10

30

45

9

69

77

127

150

**SAMPLE OUTPUT**

8

42

33

4

27

19

1

222

**Time Limit:**2,0 sec(s) for each input file.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Visual Basic

<https://www.hackerearth.com/fr/practice/math/number-theory/basic-number-theory-1/practice-problems/algorithm/can-you-guess/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

int q = int.Parse(Console.ReadLine());

while (q-- > 0)

{

int n = int.Parse(Console.ReadLine());

int sum = 0;

for (int i = 1; i <= Math.Sqrt(n); i++)

{

if (i != n && n % i == 0)

{

//div.Add(i);

sum += i;

if (i != 1 && n / i != i)

{

//div.Add(n / i);

sum += (n / i);

}

}

}

Console.WriteLine(sum);

}

Console.ReadLine();

}

}

}