acm 2008

Greater New York Programming Contest

St. Joseph's College
Patchogue, NY

event sponsors

D · Recursively Palindromic Partitions

A *partition* of a positive integer ***N*** is a sequence of integers which sum to ***N***, usually written with plus signs between the numbers of the partition. For example

`15 = 1+2+3+4+5 = 1+2+1+7+1+2+1`

A partition is *palindromic* if it reads the same forward and backward. The first partition in the example is *not* palindromic while the second is. If a partition containing ***m*** integers is palindromic, its left half is the first **`floor(m/2)`** integers and its right half is the last **`floor(m/2)`** integers (which must be the reverse of the left half. (**`floor(x)`** is the greatest integer less than or equal to ***x***.)

A partition is *recursively palindromic* if it is palindromic and its left half is recursively palindromic or empty. Note that every integer has at least two recursively palindromic partitions one consisting of all ones and a second consisting of the integer itself. The second example above is also recursively palindromic.

For example, the recursively palindromic partitions of 7 are:

`7, 1+5+1, 2+3+2, 1+1+3+1+1, 3+1+3, 1+1+1+1+1+1+1`

Write a program which takes as input an integer ***N*** and outputs the number of recursively palindromic partitions of ***N***.

Input

The first line of input contains a single integer ***N***, (1 ≤ ***N*** ≤ 1000) which is the number of data sets that follow. Each data set consists of a single line of input containing a single positive integer for which the number of recursively palindromic partitions is to be found.

Output

For each data set, you should generate one line of output with the following values: The data set number as a decimal integer (start counting at one), a space and the number of recursively palindromic partitions of the input value.

Sample Input	Sample Output

`3`
`4`
`7`
`20`
	`1 4`
	`2 6`
	`3 60`