MaxHailperin / MCS-394-S2012 https://gustavus.edu/+max/courses/S2012/MCS-394/

create pull request

student work from the problem-solving course

Clone this repository (size: 85.7 KB): HTTPS / **SSH** / SourceTree

```
$ git clone git@bitbucket.org:MaxHailperin/mcs-394-s2012.git
```

```
git clone https://MaxHailperin@bitbucket.org/MaxHailperin/mcs-394-s2012.git/wiki
```

Matt Panciera of the Gustavus Classics Department wrote the following email:

```
The Greeks liked to play around with names and they sometimes did something called isopsephism where they would
represent a person's name with a number. For example, "I love a girl whose number is 51." Since their letters each had
a numerical value (alpha = 1, beta =2) 51 would be a name where the sum totals of the letters came to 51. I would love
to generate the possible Greek names represented by these numbers. Obviously there are many ways to arrive a sum total
of 51, though at least we would be restricted by the range of length that most Greek names are. Are there any
calculators on the web that would allow me to generate all possibilities for a particular number?

My favorite example of this by the way is the emperor Nero's name in Greek (NERWN) which comes to the sum total of
1005. The ancient biographer Suetonius notes that some people noticed, soon after Nero's death, that another way to
arrive at 1005 were the letters in the sentence idian mhtera apekteine which means "he killed his own mother", which
he did in fact do.
```

Matt mentions that the letter values start with 1 and 2 for alpha and beta. He doesn't mention how it continues from there. An extended 27-letter version of the Greek alphabet is used with the values being 1, 2, ..., 9, 10, 20, ..., 90, 100, 200, ..., 900.

Matt wants to "generate all possibilities," but given that there may be an awful lot of them, we'll settle for a count of how many exist. Therefore, we will ask how many strings add up to a target number T (such as 51 or 1005) given that the length of the string is limited to at most L. Your program needs to work for values of T as large as 10000 and values of L as large as 20.

Input: There will be multiple test cases. Each case has the two positive integers T and L on a line with a space between them. After the last case is a line with two 0 values.

Output: For each case, output the case number and the resulting value, as shown in this example.

Sample input:

```
32 2
1005 2
1005 3
0 0
```

Sample output:

```
Case 1: 2
Case 2: 0
Case 3: 27
```

This revision is from 2012-05-04 01:30