All of you know a bit or two about hashing. It involves mapping an element into a numerical value using some mathematical function. In this problem we will consider a very 'simple minded hashing'. It involves assigning numerical value to the alphabets and summing these values of the characters.

For example, the string "acm" is mapped to $1 + 3 + 13 = 17$. Unfortunately, this method does not give one-to-one mapping. The string "adl" also maps to 17 $(1 + 4 + 12)$. This is called collision.

In this problem you will have to find the number of strings of length L, which maps to an integer S, using the above hash function. You have to consider strings that have only lowercase letters in strictly ascending order.

Suppose $L = 3$ and $S = 10$, there are 4 such strings.

1. abg
2. acf
3. ade
4. bce

"agb" also produces 10 but the letters are not strictly in ascending order.
"bh" also produces 10 but it has 2 letters.

Input

There will be several cases. Each case consists of 2 integers L and S $(0 < L, S < 10000)$. Input is terminated with 2 zeros.

Output

For each case, output '`Case#:`' where # is replaced by case number. Then output the result. Follow the sample for exact format. The result will fit in 32 signed integers.

Sample Input

```
3 10
2 3
0 0
```

Sample Output

```
Case 1: 4
Case 2: 1
```