Problem 3-Roman Order

Designed by Dr. Vinay Madanbhavi Shashidhar

When Marcus Aurelius looks at a list of numbers, it bothers him when numbers are out of order, but it doesn't bother him a whole lot. Marcus likes to count inversions in a list of numbers. For example, consider the list [5,4,3,2,1]. Every single pair of integers, adjacent or non-adjacent, is out of order. <5,4>, <4,2>, they all are. Since there are ten pairs of integers, there are ten inversions. But Marcus is a forgiving sort. He figures that that if the larger number is no larger than c times the smaller in any given pair, then he really doesn't care what order they're in. If c were 2 in the example above, then <5,2>, <5,1>, <4,1>, and <3,1> are the only pairs in which the pairs are out of order *and* the larger number is more than 2 times the smaller number. Marcus calls these "significant inversions." This is what you need to compute. HINT: You'll never get it by trying every possible inversion, counting them one at a time. It will take too long. You will need to count them in clumps.

INPUT SPECIFICATION. Each input case consists of the following: N followed by **<*EOLN*>,** where N is the size of the list; N is guaranteed to be no larger than 200000. The factor c followed by **<*EOLN*>** where c is the factor by which the larger number is allowed to exceed the smaller. The list containing N numbers, separated by one space and followed by **<*EOLN*>.** All numbers are normal 32-bit integers and there are no negative numbers anywhere in the input. A zero followed by **<*EOLN*>** will follow the last input case.

OUTPUT SPECIFICATION. The output cases should appear in the same order as their corresponding input cases. Each output case should be the following: "Case c: There are i inversion(s)." where c is the case number and is the number of significant inversions. Two **<*EOLN*>** characters should follow each output case.

SAMPLE INPUT

```
5<EOLN>
1<EOLN>
5·4·3·2·1<EOLN>
5<EOLN>
2<EOLN>
5·4·3·2·1<EOLN>
6<EOLN>
5<EOLN>
3·6·2·1·4·5<EOLN>
0<EOLN>
<EOF>
```

SAMPLE OUTPUT

```
Case·1:·There·are·10·inversion(s).<EOLN>
<EOLN>
Case·2:·There·are·4·inversion(s).<EOLN>
<EOLN>
Case·3:·There·are·1·inversion(s).<EOLN>
<EOLN>
<EOF>
```