In this problem we will deal with multisets and sequences. The definitions given here can differ (slightly) from those you know or are generally accepted, so please read them carefully.

A *multiset* is a collection of, possibly not unique, natural numbers, called the elements of the multiset. We denote a multiset by listing the elements in non-decreasing order, separated by commas and enclosed in brackets. The number of elements in a multiset is called its size. The size of a multiset can be zero, which means we have an empty multiset. Possible multisets are: (), (1), (-18,-17,-16) and (1,3,3,3,4,5,5,6).

A *sequence* is a multiset where the elements are in a particular order. We denote a sequence by listing the elements in the specific order, separated by commas and enclosed in braces. Like a multiset, a sequence has a size and can be empty. Possible sequences are: {}, {1}, {1,2,3}, {2,3,1} and {25,-18,25,25,7}.

Some sequences can be compared with each other. If two sequences have the same size and all elements are the same and in the same order, we call them *equal*. If two sequences have the same size and all elements are the same, but possibly in a different order, we call them *similar*. If two sequences have the same size but possibly contain different elements, we call them *comparable*. We can negate these properties by using the terms *unequal*, *dissimilar* and *incomparable* respectively. {1,2,2,3} and {1,2,2,3} are equal, similar and comparable; {6,3,1,1} and {1,3,1,6} are unequal, similar and comparable; {1,2,3} and {-988,7,-10} are unequal, dissimilar and comparable. Two sequences that have different sizes are unequal, dissimilar and incomparable.

Two sequences that are unequal but comparable can be ordered. We do this by scanning both sequences element by element, from left to right, until we encounter a difference between the sequences. If the differing element in one sequence is smaller than its counter-element in the other sequence, the first sequence is said to be smaller than the second. Using this ordering, we can sort any collection of mutualy unequal but comparable sequences and determine the *rank* of a particular sequence within that collection. The smallest sequence is said to have rank 0, the next smallest rank 1, etc. The collection of the following sequences: {7,-1,8}, {1,1,1}, {-9,0,-3}, {1,1,0} can be sorted: {-9,0,-3}, {1,1,0}, {1,1,1}, {7,-1,8}. So {-9,0,-3} has rank 0 within the collection, {1,1,0} has rank 1, etc..

We will consider five kinds of operations on multisets and sequences in the form of commands:

1. We can **degrade** a sequence to form a multiset with the same elements.

   The command to degrade a sequence is: '**degrade** *sequence*'. Example: "degrade {6,3,-1,4,-1}" gives "(-1,-1,3,4,6)".

2. We can **promote** a multiset to form a sequence with the same elements. Because one multiset can form a collection of similar sequences, we also need the rank of the sequence we want.

   The command to promote a multiset is: '**promote** *multiset rank*'. Example: "promote (4,6,8) 3" gives "{6,8,4}".

3. We can determine the **rank** of a sequence within the collection of sequences that can be degraded to the same multiset.

   The command to rank a sequence is: '**rank** *sequence*'. Example: "rank {8,6,9,6}" gives "7".

4. We can **derive** a sequence of a certain size from a multiset, if that size is smaller than or equal to the size of the multiset. Because a collection of comparable sequences can be derived from a multiset, we also need the rank of the sequence we want.

   The command to derive a sequence from a multiset is: '**derive** *multiset size rank*'. Example: "derive (1,1,2,2,3) 3 15" gives "{3,1,2}".

5. Finaly we can **find** the rank of a sequence within the collection of comparable sequences of a certain size that can be derived from the same multiset.

   The command to find a sequence is: '**find** *sequence multiset*'. Example: "find {5,8} (0,0,1,1,2,2,3,3, gives "58". Notice that the size is not given explicitly in the command because it is implicitly given with the sequence parameter.

Input

The input will consist of a series of commands as defined above, each on a line by itself and without the surrounding quotes. Multisets and sequences will have a minimum of 0 and a maximum of 20 elements. Elements are 32-bit signed integers. All commands will be valid. One space character will be used to separate the command from its parameters and one space character wil be used between parameters. There will be no spaces within multisets or sequences and no leading or trailing spaces. Command names will be written in lowercase letters. The end of the input is indicated by the word '**end**' on a line by itself. This line produces no output.

Output

For each command in the input print the result, either a number, multiset or sequence as defined above on a line by itself and without the surrounding quotes. The output should not contain any space characters. The elements of a multiset should be printed in non-decreasing order.

Sample Input

```
degrade {6,3,-1,4,-1}
promote (4,6,8) 3
rank {8,6,9,6}
derive (1,1,2,2,3) 3 15
find {5,8} (0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9)
end
```

Sample Output

```
(-1,-1,3,4,6)
{6,8,4}
7
{3,1,2}
58
```