Let n and k be numbers with $n > 0$ and $k \geq 0$. A *configuration* of the n-k-puzzle is an n-tuple with elements in the range $-k \ldots k$ such that their sum is zero. Configurations are considered equivalent when they can be obtained from each other by (a) cyclic permutation of the tuple over one or more positions, (b) reversal of the tuple, (c) sign reversal of all elements, or (d) combinations of (a), (b), and (c). Equivalence classes are called *patterns*.

For instance, (0, 1, 1, -2) is a configuration of the 4-2-puzzle. Some equivalent configurations are: (a) (1, -2, 0, 1), (b) (-2, 1, 1, 0), (c) (0, -1, -1, 2), and (d) (-1, -1, 0, 2). Below is given a list of (the lexicographically largest) representatives of the 14 patterns of the 4-2-puzzle.

$(0, 0, 0, 0)$
$(1, -1, 1, -1)$
$(1, 0, -1, 0)$
$(1, 0, 0, -1)$
$(1, 1, -1, -1)$
$(2, -2, 2, -2)$
$(2, -1, 0, -1)$
$(2, -1, 1, -2)$
$(2, 0, -2, 0)$
$(2, 0, -1, -1)$
$(2, 0, 0, -2)$
$(2, 1, -2, -1)$
$(2, 1, -1, -2)$
$(2, 2, -2, -2)$

Your program computes the number of patterns for a sequence of n-k-puzzles.

Input

The input consists of a sequence of pairs of integers n and k, which are separated by a single space. Each pair appears on a single line. The input is terminated by an end-of-file. The value for $n + k$ is at most 11.

Output

For each line of the input, the output consists of an integer m, on a line by itself, representing the number of patterns for the corresponding n-k-puzzle of the input. Then m lines follow with the patterns written in lexicographical order, as shown in the Sample Output below.

The outputs of two consecutive cases will be separated by a blank line. No blank line should appear at the end of the output.

Sample Input

```
8 0
4 2
```

Sample Output

```
1
(0,0,0,0,0,0,0,0)

14
(0,0,0,0)
(1,-1,1,-1)
(1,0,-1,0)
(1,0,0,-1)
(1,1,-1,-1)
(2,-2,2,-2)
(2,-1,0,-1)
(2,-1,1,-2)
(2,0,-2,0)
(2,0,-1,-1)
(2,0,0,-2)
(2,1,-2,-1)
(2,1,-1,-2)
(2,2,-2,-2)
```