

ZBROJ

Because the result doesn't necessarily fit in a 32-bit integer type, we'll build it as a string during calculation. The official source codes in different programming languages show various approaches to implementing the solution:

- In C, the numbers are read as strings, addition is performed right to left and the result is built right to left.
- In Pascal, the numbers are read as integers. The result is again built right to left, but the code is simpler because strings are simpler to work with than in C.
- In C++, the numbers are read as strings and padded with zeros on the left to make them of equal lengths. Addition is performed left to right, which we can do because there is no carry-over, unlike in normal addition.

VLAK

For each car we keep track of:

- The number of passengers in the car;
- For each letter, how many passengers whose names start with that letter are in the car.

Based on this data we can uniquely determine which car each passenger boards.

TETRIS

Imagine that a piece is at the bottom of an empty field. The piece can then be encoded as a sequence of numbers if we write down the lowest occupied row for each column the piece occupies.

For example, piece number 5 in its four rotations gives following four sequences:

{ 1, 1, 1 }, { 1, 2 }, { 2, 1, 2 } and { 2, 1 }.

We say that sequence A corresponds to sequence B if adding some constant to all elements of A gives sequence B.

For example, sequence { 2, 1, 2 } corresponds to { 5, 4, 5 }, but not to { 4, 5, 4 } or { 5, 4 }.

Observe that a piece can be inserted at a given position only if the sequence of heights at that position corresponds to the encoding of the piece. Therefore in order to obtain the solution we simply need to check all possible positions of all rotations for the given piece and field. Note that not all pieces have 4 possible rotations. Piece 2 is the same regardless of the rotation and pieces 1, 3 and 4 have only two possible rotations.

The official source code in C++ showcases this approach, while source codes in C and Pascal use manually written conditions to check whether a piece can be placed at a certain location.

KRUG

The first step in solving this problem is to simulate the described transformation K times in order to obtain Stanko's final circle.

The reverse transformation is not unique, but there are at most two different circles that, when transformed, yield any given circle – if we fix B or W as one of the pebbles, the rest are determined uniquely.

Therefore there are at most 2^K different possible starting circles. Since K is small enough we can check all of these and remove any duplicates.