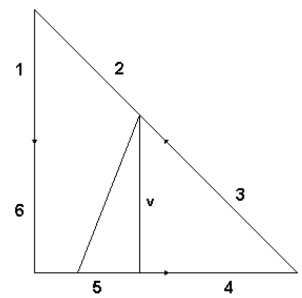
BUKA

For multiplication, the number of zeroes in the result is the sum of the numbers of zeroes in both numbers. For addition, only a single digit in the larger number will change: a zero into a one, or the leading one into a two.

BAZEN

There are six distinct cases, depending on where the given point is – in which of the six labeled parts in the figure. In each of these cases, splitting the triangle in half yields a smaller triangle. We know either the base or height of this smaller triangle from the coordinates of the given point. We can calculate the other length from the known area of the triangle. Finally, from these two numbers we can calculate the coordinates. It is important to note that all points on the hypothenuse have $x+y = 250$.



NERED

Cubes that are on top of other cubes will surely need to be moved so we can add them to our output immediately. Now we need to find the rectangle that contains the most cubes, with an area equal to the total number of cubes. This rectangle can be found by considering all possible rectangles.

CUSKIJA

For simplicity, assume all input numbers are 0, 1 or 2. This does not change the result since we only need the modulus by 3.

Obviously two zeroes or a one and two cannot be adjacent. If the total number of zeroes is two or more greater than the number of ones and twos, then we can't avoid placing two zeroes together and there is no sequence. Similarly, if there are no zeroes but there are ones and twos, then we would have to place a one and two together.

Once we know it is possible, we can construct the sequence by inserting groups consisting only of ones or twos between zeroes.

DOSTAVA

The traffic grid can be modeled with a graph consisting of $2 \cdot R$ vertices, one for each of two endpoints in a row. Every vertex (except the ones in the first and last rows) is connected to three other vertices – the ones above and below it, as well as the other endpoint in the same row. Knowing the shortest paths between every pair of vertices in this graph allows us to easily calculate the shortest paths in the original grid. There are four cases – we can choose to start going left or right, and to approach the destination from the left or the right.

The all-pairs shortest paths problem can be easily solved by the Floyd-Warshall algorithm in $O(R^3)$ complexity, enough for 70% of points.

For full points, we can implement Dijkstra's algorithm or a dynamic programming solution that takes advantage of the special properties of the graph. First note that the shortest path between the endpoints of two rows A and B may go through rows outside the interval [A, B], which introduces cycles in our state space, a obstacle when using dynamic programming. This can be solved by precalculating the shortest path from one endpoint of the row to the other. After this we can use dynamic programming to populate the shortest paths table. The overall complexity is $O(R^2)$.

SLICICE

The problem can be modeled with a flow network, the maximum flow through the network being the answer. The vertices in the network are the source, sink, purchases and children. The number of purchases is half the number of cards.

The source is connected to all purchases with links of capacity two. Known purchases are connected only to the two children who went, again with links of capacity two. Unknown purchases are connected to all children. Finally, we connect all children to the sink, the capacity of links equal to the number of cards they have in the end. The maximum flow through the network tells us how the cards were distributed.