

## INSTRUKCIJE

Using two nested for-loops we generate the sequence consisting of one number 1, two numbers 2, three numbers 3 etc.

After that, the sum of numbers in indices A and B is easy to find using another for loop.

## PJESMA

Assign to each word in the title of the song a flag telling us whether the word has occurred in the lyrics. For every word in the lyrics we check if it is part of the title, and if it is and it has not occurred so far, set the flag and increase the counter.

Once the counter is  $N/2$  or larger, stop the algorithm and output the index of the current word.

## LAGNO

For every empty square, calculate the number of white stones which can be converted to black if we put a piece there. To achieve this, prepare a counter and, for each of the 8 directions, move in that direction while the pieces are white. If there is a black piece after the white pieces, then those white pieces will be converted and we should count them.

The value to output is the biggest number of pieces we can convert by placing a black piece in one of the empty squares.

## NIKOLA

We model the problem as a shortest path problem in a graph in which the state is an ordered pair (square, jumplen), where "square" is the square at which Nikola is located and "jumplen" is the length of the preceding jump.

From state (square, jumplen) Nikola can move to state (square-jumplen, jumplen) moving backwards, or to state (square+jumplen+1, jumplen+1) moving forwards, assuming these jumps don't move him out of the playing area.

Notice that the graph is acyclic (meaning that there is no way to visit a state twice while traversing the graph), because forward jumps increase the "jumplen" parameter. Because of this, the length of the shortest path can be calculated with dynamic programming.

Let the function  $\text{opt}(\text{square}, \text{jumplen})$  represent the smallest cost for Nikola to get from square "square" to square N, if the preceding jump was of length "jumplen".

Here's how to calculate the value of the function:

- For  $\text{square} < 1$  or  $\text{square} > N$ ,  $\text{opt}(\text{square}, \text{jumplen}) = \infty$ ;
- For  $\text{square} = N$ ,  $\text{opt}(\text{square}, \text{jumplen}) = \text{fee}[N]$ ;
- Otherwise,  $\text{opt}(\text{square}, \text{jumplen}) = \text{fee}[\text{square}] + \min\{ \text{opt}(\text{square}-\text{jumplen}, \text{jumplen}), \text{opt}(\text{square}+\text{jumplen}+1, \text{jumplen}+1) \}$

The solution is then  $\text{opt}(2, 1)$ .