

# Tasks of the third stage of the Logia18 contest

## - the subject IT competition

### for junior high school students of the Mazowieckie Voivodeship

#### March 7, 2018

#### Task 1 (way).

In Turtland, the roads are unidirectional, with at most one of them leaving the village road. The cost of a change from each town to another connected directly with it is 1. Ania sets off the city A and Zbyszek With .

Write a one-parameter `road` function , the result of which will be the minimum cost needed for Ania and Zbyszek to meet in one place or -1 when they can not meet. parameter the function is a list of two-element lists describing connections between cities, where the first element pairs is the identifier of the place where the road begins and the second - where it ends directly. The city identifiers are capital letters of the Latin alphabet. The maximum length of the list is 26.

*Examples:*

in the Logo - the result <code>route [[AB] [ZD]]</code> is -1 ,	in Logo - the result of the route [[AB] [GH] <code>[BC] [ZC]]</code> is 3 ,	in Logo - the result of the route [[ZD] [AB] <code>[FB] [BC] [CD] [DF]]</code> is 4 ,
in Python - the result <code>road ([[ 'A', 'B'] [ 'Z', 'D']])</code> is -1 .	in Python - the result of the route ([ 'A', 'B'], <code>way ([ 'Z', 'D'], [ 'A', 'B'], [ 'F', 'B'],</code> <code>[ 'G', 'H'], [ 'B', 'C'], [ 'Z', 'C']])</code> is 3 ,	in Python - the result <code>way ([ 'Z', 'D'], [ 'A', 'B'], [ 'F', 'B'],</code> <code>[ 'B', 'C'], [ 'C', 'D'], [ 'D', 'F']])</code> is 4 .

#### Task 2 (neon).

In Turtland, they prepare a neon for hanging on two pillars. The poles are standing on a square grid, and the distance between two adjacent posts in a row and column is 2. The neon must be hung in one row or column. The marketing department makes the choice of columns dependent on the rating defined as the sum of them height and distance between them.

Write a one-parameter `neon` function , which results in the highest possible rating. The parameter is list of column height lists. The rows are described by further sub-lists. In each row and column is from 2 up to 500 poles. Make sure that you do not have to wait too long for the result of the function.

*Examples:*

in the Logo - the result of the neon `[[1 9 2] [3 8 3] [2 1 1]]` is 19 ( $19 = 9 + 8 + 2$ ),  
 the result of the neon `[[1 2 1 2] [7 1 7 1] [1 1 1 1] [3 3 3 3]]` is 18 ( $18 = 7 + 7 + 2 \cdot 2$ ),  
 in Python - the result of neon `([[1, 9, 2], [3, 8, 3], [2, 1, 1]])` is 19 ( $19 = 9 + 8 + 2$ ),  
 the result of neon `(([[1,2,1,2], [7,1,7,1], [1,1,1,1] [3,3,3,3]])` is 18 ( $18 = 7 + 7 + 2 \cdot 2$ ).

#### Task 3 (pages).

Marek marked the websites he visited with the next lower case letters of the Latin alphabet. He visited each of them exactly once, but does not remember exactly in which order. For some sites he remembers which he visited before and which he later visited (eg that he visited a page before c and a page d before b ).

Write a two-parameter `page` function . The first parameter determines the number of pages visited and can take values from 2 to 8. The second is a list (no more than 30) of two-letter words determining the order in which certain pages were visited (eg the word `ab` means that page a was visited earlier than b ). The value of the function is the number of possible order of visits to all pages.

*Examples:*

in Logo - the result of `page 2 [ba]` is 1 (possible order of visiting: ba ), the result of `page 4 [ab ac`

ad db] is 3 (possible visit order: acdb, adbc, adcb.)  
 in Python - the result of the page (2, ["ba"]) is 1 (possible order of visiting: ba), the result  
 page (4, ["ab", "ac", "ad", "db"]) is 3 (possible order of visits: acdb, adbc, adcb).

## Task 4 (dividers).

Write a two-parameter `divisors` function with total positive parameters `a` and `b`, which  
 the result is the number of these integers from the range  $< a, b >$  (i.e. not smaller than `a` and not larger  
 than `b`, we assume that  $a \leq b \leq 1000000$ ), which have only three different natural divisors.

*Examples:*

in Logo - the result of `divisors 2 6` is 1 (the above property has only 4), the result `divisors 80,000 90000` is 2 ,

in Python - the result of `divisors (2, 6)` is 1 (the above property has only 4), the result

`divisors (80000,90000)` is 2 .