

# Algorithms and labyrinths. Solutions.

A. Belov-Kanel, I. Gazhur, I. Ivanov-Pogodaev, A. Malistov

**A11.** Suppose that the robots can see other robot's flags. Let our robots use standard algorithms for tape exploring, one for horizontal line and other for vertical line. Also, if robot see the other robot's flag, he stops. Suppose that vertical line intersect horizontal one in cell  $O$ . If the distances of the each robot to  $O$  are the same at the beginning then the robots will meet in  $O$ . Now, suppose that robot's  $A$  distance is more than  $B$  one. In this case, then  $A$  comes to  $O$ , he will be between the other robot's flags. According his algorithm,  $A$  drop his flag to  $O$ . Using the fact that our robots are synchronized, we can conclude that during the time that takes  $A$  to go to his second flag and return to  $O$ -flag, robot  $B$  will pass between his flags and cross  $O$  cell. So he will stop there. In some tome  $A$  robot will come to  $O$  too.

This problem can be solved for the case that the robots could not see the other robot's flags. You can find this solution.

## sections B and C.

The solutions of B and C problems are algorithms designed in special application VillagerLife (attached).

Attached materials contain several folders, one folder for each problem. There are some test maps (named test1.vlm, test2.vlm etc) in each folder. Also there are some algorithms solving that problem. We have attached the algorithm written by participants of the conference. The name of each algorithm contains the names of authors. If there are no names, the algorithms was designed by Jury.

Some interesting algorithms were designed by Dmitry Belikov and Konstantin Khadaev, Danila Baigushev, and Brian Chen, which was the only participant that solved the complicated problem C10.

Actually, problems C5-C7 are step tasks for the main problem C8, so we did not include tests and algorithms for them.

We used two packs of tests for C8 problem: for two various AI robots. Also, we attached the algorithm designed by Dmitry Belikov and Konstantin Khadaev.