**Draft version (Emil Kelevedjiev)**

**Distant competition in informatics by the medium of "Computer" Bulgarian Magazine**

On the pages of the "Computer" Bulgarian Magazine (issued by the New Technik Publishing), the distant competition in Informatics has being maintained, starting from 1987. We may point out the following main features of this competition:

– There are given about 6 tasks each year;
– The tasks have algorithmic nature, similar to those given at International Olympiad in Informatics;
– The student has plenty of time to solve the proposed problem – typically a month;
– Student's solution should contain not only a programming code, but also some explanations;
– The evaluation uses test examples;
– The evaluation uses also examination of student's explanations;
– After the end of the period, when the participants should submit their solutions, the author of the proposed problem publishes the detailed description of his own solution, accompanying it with explanations and discussions of students' works.

We can summarize the topics in algorithms, which were appeared in the tasks of the mentioned competition, given during the years (at some topics there are written in parentheses the corresponding numbers of the selected tasks from the Appendix):

By Subject:

- Number Theory (4-1990);
- Combinatorics;
- Sorting and searching (3-1997);
- Graphs (3-1989, 3-1989);
- Computational Geometry (3-1991);
- Games Theory (2-1990)

By Method:

- Basic Technique: Iteration and Recursion (3-1989);
- Basic Technique: Data structures;
- Divide and conquer;
- Dynamic Programming (2-1998);
- Greedy Algorithms;
- Exhaustive search, e.g. backtracking (4-1990, 4--1994, 1-1998);
- Numerical-like methods problems (5-1999)

The tasks given on this distant competition can serve also as a source for learning (provided that the published author's solution is very detailed and contains some extensions in the theory), we give in the Appendix a short list of several tasks, taken from different years. For some of them, their explanations could have a form of short

introductions (to the Theory of Graph – for 3-1989, or, to the Game Theory -- for 2--1990, etc.).

**Appendix**: Selected tasks given at the Competition of "Computer" Bulgarian Magazine.

**Task 3-1989**. Given a set of points, some points connected by segments, write a program to print out a route (if there exists), that begins and ends at some of the given points, and passes through every given segment exactly once.

**Task 8--1989**. Given a maze, represented as a rectangular table, each cell containing 0 or 1, write a program to find out how somebody can go from one cell to another, travelling only through zero cells in such a way, that after visiting a cell, he/she is allowed to continue through a) Any one of the four neighbors (on the left, right, up, or down); b) Any one of all the eight possible neighbors.

**Task 2-1990**. Two people play a game in which a long rope is being cut consequently by the first and the second player. The person losing is the one who cuts a piece of the rope, smaller than 1 centimeter. Write a program that makes wining moves for the first player.

**Task 4-1990**. Seats in a hall are arranged in a rectangular array with M lines with N seats in each line. There is a person sitting on each seat. His/her neighbors are those, sitting on the left, right, in front and behind of him/her. Every person has up to 3 enemies from among the other people in the hall. Write a program to interchange the people's places, so that there will be no enemies who are neighbors.

**Task 3-1991**. There is given a simple (having no self-crossing of its sides, but not necessary convex) polygon in the plane, with coordinates of the vertices. The vertices are numbered consecutively according to a chosen walk direction of polygon's boundary. Write a program that determines whether a given point lies inside the polygon.

**Task 4-1994**. There is given a rectangle with integer lengths A and B of its sides. Write a program, that determines whether the rectangle can be covered entirely (with no overlapping, no remaining uncovered parts and no stick out's) by N small rectangles, each having given integer lengths A(i) and B(i) of its sides.

**Task 3-1997**. There are given N stones, numbered from 1 to N. It is known that the majority, i.e. more than the half of all the stones have the same weight. There is a device which can determine whether two stones have the same weight, but the device cannot determine which is heavier in case they don't have the same weight. Write a program that clarifies how to use the device to find out a stone that belongs to the majority. It is allowed to use the device no more than N times to solve this problem.

**Task 1-1998**. In a town, there are several exchange offices. Each office performs exchanges of several foreign currencies according to its own given table of exchange rates, separately for selling and buying. A man has 100 dollars. He wants to play smart - choosing a sequence of offices and making an interesting way of selling and

buying, he wants to end up with more than 100 dollars. Write a program to find out such a sequence of exchanges, if it is possible.

**Task 2-1998**. We have a convex polygon made of a thin sheet of metal. Its vertices are numbered from 1 to N around polygon's contour. We make straight-line cuttings connecting arbitrary two vertices using scissors. Cuttings are made in a sequence, such that at the end of the process, all of the remaining parts of the initial polygon become triangles. Defining blunt of scissors as the sum of lengths of all the cuttings, write a program to find out the sequence of cuttings, giving minimal blunt of the scissors.

**Task 5-1999**. In the plane, there are given coordinates of M red and N blue points. Write a program that finds out a straight line, which has an equal quantity of red points lying on both parts of the plane, relatively to the straight line. The same should be true for the blue points.