INTERNATIONAL OLYMPIAD IN INFORMATICS

Pravetz, May 16 - 19, 1989
BULGARIA

PROCEEDINGS

OF THE

INTERNATIONAL OLYMPIAD

IN

INFORMATICS

Pravetz, May 16 - 19, 1989

EDITED BY:
Petar S. Kenderov,
Nelly M. Maneva

UNION OF THE MATEMATICIANS IN BULGARIA

Sofia 1989

C O N T E N T S

ОТПЕЧАТАНО В ПЕЧАТНАТА БАЗА
НА С Н Р Б
БУЛ."ЛЕНИН" № 73
СОФИЯ,1989

1. INTERNATIONAL OLYMPIAD IN INFORMATICS

PRELIMINARIES AND GENERAL INFORMATION

An International Olympiad in Informatics (further on abbreviated as IOI) for secondary school students was held in Pravetz, Bulgaria in May 16–19, 1989.

The idea to organize an International Olympiad in Informatics (IOI) was brought into the focus of the attention of the 24–th session of the General Conference of UNESCO held in Paris (October – November 1987) by *Professor Blagovest Sendov, a member of the Bulgarian* delegation. Debates resulted in mentioning IOI in the Fifth Main Programme of the UNESCO Plan for 1988–89 (section 05 215) together with other Olympiads (in Mathematics and Natural Sciences). Under a contract with the UNESCO Division of Science, Technical and Environmental Education Bulgaria took the obligation to realize the first IOI in 1989.

IOI was organized by the Bulgarian Ministry of Culture, Science and Education with the active cooperation of the Union of the Bulgarian Mathematicians, the Institute of Mathematics of the Bulgarian Academy of Sciences and the local authorities in Pravetz.

Thirteen countries sent their students to compete in IOI. These were: Cuba, Czechoslovakia, Federal Republic of Germany, German Democratic Republic, Greece, Hungary, Peoples Republic of China, Poland, USSR, Vietnam, Yugoslavia, Zimbabwe and Bulgaria. According to the rules of IOI (Section 2) a team should consist of no more than three school students accompanied by an elder person – the leader of the team. The Hungarian and Yugoslavian teams had two students each. Bulgaria participated in IOI with two teams and the Soviet Union was represented by three teams. Thus 16 teams including 46 students took part in this IOI.

On Wednesday morning (May 17, 1989) the *International Jury itself*, namely the Chairman, the Deputy Chairman and the team leaders *gathered and discussed* the six problems prepared by the Coordinating Commission in advance. Those problems were based on some suggestions made by the team leaders before the IOI. The International Jury selected one problem (originally proposed by China) which was then translated into English and Russian. The team leaders translated the problem into the respective languages understandable for their students.

The *competition started in the afternoon* on that day (May 17th). The translated versions of the problems were given to the students who had four hours at their disposal to work on them. During the first 30 minutes the students had the right to ask the Jury (being still apart) some questions (in a written form) concerning the formulation of the problem. The lists comprising the questions were translated into English by the respective team leaders and after that the Jury decided what their answers should be.

Each student had a microcomputer at his/her disposal. As desired by the participants most computers were IBM PC compatible. They were provided by the organizers. The team from the Federal Republic of Germany and one of the teams from the Soviet Union participated in IOI with their own equipment.

At the end of the competition each team leader accompanied by a member of the Coordinating Commission collected the problem solutions from the students of the respective team. The work of each student (the final version of the solution) was copied on two floppy disks. One remained with the team leader and the other stayed with the Coordinating Commission. Similarly, two identical printouts comprising the source texts of each student's program were collected and divided among the team leaders and the Coordinating Commission members. The program of each student was also run with some unified test examples (Section 8). Further on the Jury members and the students did not separate any more which *helped the team leaders considerably in the process of checking and* preliminary marking their students' works.

On the next day (May 18, 1989) the Jury members had to coordinate the number of points to be assigned to the students' work. Meanwhile the students participated in an entertainment programme. Professor Iltscho Dimitrov, the Minister of Education gave a reception for the participants of the IOI in the evening.

The final marking of the papers and the decision which of the students should be given the first, the second or the third prizes were made on Friday morning (May 19, 1989). The maximal possible number of points given for a complete solution of the problem was 100 (see distribution for the different subproblems in Section 8). The First prize (G) was awarded to the following students: Teodor Tonchev (BG-2), Markus Kuhn (FRG), Emanuil Todorov (BG-1), Andrius Cepaitis (USSR), Igor Maly (CS), Daniel Szabo (H) who got more than 90 points.

The Second prize(S) was given to five students who got 80-90 points. The Third prize (B) was awarded to seven students whose number of points attributed amounted to the range of 60 - 80 (Section 8).

Two stimulative prizes were also awarded. The winners were respectively: Alexei Kolybin (USSR) - the youngest participant, 13 years old and Anita Laloo (Zimbabwe) - the only girl among the participants.

At the closing ceremony the prizes were given to the winners by Vladimir Zhivkov, President of Lyudmila Zhivkova International Foundation.

All the students were given an honorary diploma certifying their participation in IOI.

The expenses of the team participants and the team leaders for their stay in Bulgaria were covered by the organizers.

The IOI future was discussed at the Jury's sessions. Most of the participating countries emphasized the necessity to organize IOI every year. The representative of Greece announced that it is possible to organize IOI in Greece (in 1991) and the representative of the Federal Republic of Germany expressed the willingness of his country to host IOI in 1992.

Quite recently (in August 1989) a piece of good news came from the USSR that the next IOI would be organized in Byelorussia (Minsk, July 15–21, 1990).

At a suggestion of A. Pokrovsky (UNESCO, Division of Science, Technical and Environmental Edication) and in order to keep alive the spirit of IOI a working group was created, its main task being to improve the exchange of information among the participating countries and to discuss new ideas for organizing future IOIs. The members of this group are listed in Section 10.

Many people contributed to the IOI organization in different ways and at various stages. The work of International Jury was supported by the software system created by P. Azalov and V. Dimitrov. In the hands of I. Nenova and V. Dimitrov this system served perfectly all the information needs of IOI – starting with the registration of participants and ending with the ranking with respect to the results obtained in the competition. Some outputs of this system could be found in Section 8. Section 7 was compiled by P. Azalov and K. Manev.

IOI was the third international informatics competition which had been organized in Bulgaria. The first one named "Open Competition on Programming" took place in Sofia, May 17–19, 1987, as an integral part of the second international conference "Children in the Information Age" (Appendix 1). The second competition was held in Varna, October 5–8, 1988 and it was for technical school students (Appendix 2).

IOI took the place of the Open Competition on Programming which was to be held just before the Third International Conference "Children in the Information Age" (Sofia, May 20–23, 1989).

2. REGULATIONS OF THE INTERNATIONAL OLYMPIAD IN INFORMATICS

1. GENERAL INFORMATION ABOUT THE IOI

1.1. IOI is an informatics competition for school students who are not older than 19 years on May 16, 1989 (i.e. who were born after May 16, 1970).

1.2. Each country participates in IOI with a team of three (or less) school students accompanied by an elder person – the leader of the team. It is preferable to have both boys and girls in the team.

1.3. The IOI consists in solving one problem by using personal computer (PC) and it is carried out in one day within four hours.

1.4. The problem will be selected by the International Jury (see 2.1.) on the basis of the problems, provided in advance by the participating countries.

1.5. The problem will be independent of the PC hardware and high level programming languages can be used for its solution (e.g.BASIC, LOGO, PASCAL, etc.). The problem will be of algorithmic nature and no specialized software packages etc. will be necessary to solve it.

1.6. The complete solution of each problem includes:

a) An informal and block–diagram description and argumentation of the algorithm on the basis of which the program is written;

b) The source text and results of the program execution, obtained by the microcomputer (2 copies);

c) The floppy disk on which the final version of the program is stored, in the source language and eventually in executable code (2 copies).

1.7. After the end of the competition one copy of the materials mentioned in 1.6. b) and c) is given to a representative of the coordination commission (see 2.2.). The other copy of these materials as well as everything connected with 1.6. a) is kept by the team leader.

1.8. The evaluation of solution will be made by the leader of the respective team. The final marking will be made by the coordination commission and the International Jury.

1.9. During the competition every student can use his/her own microcomputer he/she is familiar with. Every student should be able to input, run and transfer programs from one data carrier to another. He/she has to possess the required number of floppy disks, which will be used to store software, providing the programming environment, as well as two empty floppy-disks.

On preliminary request, the organizers can provide for the competition a number of

- APPLE II+ compatible, or

- IBM PC /XT/AT/ compatible computers.

1.10. Working languages for IOI:

a) Students can use their native language:

b) The team leader must understand the languages used by his/her students and he has be able to speak and understand at least one of the official languages of the competition - Russian and English.

1.11. The organizers will use the experience from the International Mathematical Olympiads to conduct the IOI.

1.12. All expenses related to the stay in Bulgaria of the team and the team leader in connection with IOI will be covered by the organizers. The organizers cannot cover the costs related to the stay of any other person from the same country as well as the travel costs of the participants to and from Sofia.

2. EXECUTIVE BODIES

2.1. International Jury.
The Jury consists of

a) Chairman;

b) Deputy Chairman;

c) Leaders of all participating teams.

2.2. Coordination Commission (for checking and evaluation of the solutions). It consists of:

a) Members of the Jury;

b) 8 coordinators.

2.3. Team of interpreters.

2.4. Organizing committee.

2.5. Scientific committee.

3. PREPARATION AND CARRYING OUT THE IOI

3.1. Preliminary phase.

The Scientific Committee will choose 5 of the problems, proposed in advance by the countries represented in the IOI, and will have them translated in the working languages (English and Russian).

3.2. First phase.

a) the Jury will select one of the 5 problems proposed by the Scientific Committee and will discuss its solution, level of difficulty and way of evaluation (marking). The problem must allow evaluations of partial solutions;

The Jury formulates the selected problem in the working languages of IOI (English and Russian).

b) the formulation of the problem then will be translated by the team-leaders into the respective national languages. The translations will be checked and affirmed by the Jury;

c) the Jury will fix the final details of the IOI.

3.3. Second phase - carrying out of the competition

a) in the first half hour after the beginning of the competition the participants may ask the Jury written questions (in their national languages) concerning any unclarities in the text. The Jury will decide what to answer.

b) at the end of the competition the team-leader together with a representative of the Coordination Commission collect the problem solutions of the participants from the corresponding country (see 1.7.).

c) the leader of each team checks the solutions of his students and proposes certain marks, which will be further discussed by the coordinators. The final decision is taken by the Jury.

d) the limits of the grades for the first, second and third prizes will be defined by the Jury.

3.4. Closing phase.

a) the prizes will be awarded at a special ceremonial meeting connected with the third International conference and exhibition on "CHILDREN IN THE INFORMATION AGE" 20 - 23 May, 1989;

b) each participant will receive a certificate for participation in the IOI;

4. REMARKS

4.1. There will be an entertainment program for participants, team-leaders and organizers of IOI.

3. P R O G R A M M E

for the International Olympiad in Informatics, which will be held in the framework of the Third Conference "Children in the Information age" (Pravetz, May 16 to 20, 1989)

16.05.1989 (Tuesday)

	- Welcoming the participants in the International Olympiad in Informatics.
	- Departure for Pravetz, Accommodation in the "Pravetz" Hotel.
	- Introduction to the technical equipment at the Educational Training Centre in Pravetz.
19:00 h.	- Technical Conference of the teams' leaders and members of the International Jury.
20:00 h.	- Supper.

17.05.1989 (Wednesday)

7:00 h.	- Breakfast (for the Jury).
8:00 h.	- Breakfast (for the teams).
8:00 h.	- Session of the International Jury for defining the problems for the Olympiad. (until 12:00 h.).
9:00 h.	- *Introducing the teams to the technical equipment to* be used during the work of the Olympiad at the Educational Training Centre (until 12:00 h.).
12:30 h.	- Lunch in the Canteen of the Educational Training Centre (for the International Jury).
12:30 h.	- Lunch in the hotel (for the teams).
14.00 h.	- Holding of the Olympiad at the Educational Training Centre (until 18:00 h.).
[illegible]	- Supper in the hotel.

18.05.1989 (Thursday)

8:00 h.	– Breakfast in the hotel.
9:00 h.	– Session of the International Jury at the Educational Training Centre for evaluating the teams' work (until 13:00 h.).
9:00 h.	– Excursion to Sofia. (for the teams – until 12:00).
13:30 h.	– Lunch (in the hotel).
15:00 h.	– Evaluation of the results (for the International Jury – until 18:00 h.).
15:00 h.	– Round-trip of Pravetz. (for the teams – until 18:00 h.)
19:30 h.	– Cocktail-party on behalf of the Minister of Education (in the hotel).

19.05.1989 (Friday)

8:00 h.	– Breakfast.
9:00 h.	– Final evaluation of the results (for the International Jury – until 11:00 h.)
9:00 h.	– Free time for the teams – until 11:00 h.
11:30 h.	– Presentation ceremony of the awards from the International Olympiad in Informatics (3 First Prizes).
13:30 h.	– Lunch in the hotel.
15:00 h.	– Visit to the PC Enterprise in Pravetz. (until 17:00 h.).
19:30 h.	– Supper.

20.05.1989 (Saturday)

8:00 h.	– Breakfast.
9:00 h.	– Departure for Sofia.
	– Departure of the participants in the International Olympiad in Informatics

4. SIX PROBLEMS PRESENTED TO THE JURY OF IOI

PROBLEM 1.

Given 2 * N boxes in line side by side (N <= 5). Two adjacent boxes are empty, and the other boxes contain N - 1 symbols "A" and N - 1 symbols "B".

Example for N = 5:

```
| A | B | B | A |   |   | A | B | A | B |
```

Exchanging rule:

The content of any two adjacent non-empty boxes can be moved into the two empty ones, preserving their order.

Aim:

Obtain a configuration where all A's are placed to the left of all B's, no matter where the empty boxes are.

Problem:

Write a program that:

1. Models the exchanging of boxes, where the number of boxes and the initial state are to be input from the keyboard. Each exchange is input by the number (from 1 to N-1) of the first of the two neighboring boxes which are to be exchanged with the empty ones. The program must find the state of the boxes after the exchange and display it.

2. Given an initial state finds at least one exchanging plan, which reaches the aim (if there is such a plan). A plan includes the initial state and the intermediate states for each step.

3. Finds the minimal exchanging plan which reaches the aim.

PROBLEM 2

The floors in a building are numbered sequentially with the integers 0,1,2,...,N (N <= 15). There are K (1 <= K <= 4) lifts in the building. Lift control is centralized, and accepts two types of requests, entered by pressing buttons. External buttons (one for request to move up and one – to go down) can be found on each floor, and are common for all lifts. Internal buttons (requests to move to a given floor) are found in each lift.

Build a program to model lift group control on the following conditions:

1. There is a single lift in the building (K = 1), and it can accept a single request at a time. Any other request is accepted after completion of the first one.

2. There are several lifts in the building (K >= 1). Each of them accepts an internal request only if it is not executing an other request. The lift control device can register several incoming requests at the same time. Internal requests are fulfilled by the lift, where they were entered. The control device selects a suitable free lift to fulfill each external request.

3. Consider the same case as in 2, with the restriction that even-numbered lifts can stop at even-numbered floors, and odd-numbered lifts – at odd-numbered floors only. All lifts stop at floor zero.

4. Consider the case in 3, and suppose that there can be several pending internal requests from each lift, not just one. All internal requests are accepted and registered, no matter whether a lift is free or not.

One could accept that all lifts are synchronized, and at equal time intervals (clock ticks) each lift is located at a given floor. During the next tick, a lift could go one floor up or down, or remain at the same floor. Requests (program input) can be entered at any tick, and they are of the following types:

a) external – <floor number, direction of movement (up or down)>;

b) internal – <lift number, floor number>

Several or none requests can be entered at each tick.

At each tick the program should display information about the location of each lift.

The lifts are large enough and cannot be overloaded.

The program should control the lifts so that their behaviour shows as much "intelligence" as possible.

There should be explicit explanations of the lift control strategy.

PROBLEM 3.

Given is a group of N persons. Everybody is a friend of more than [N/2] of the others and has no more than K enemies. One of the persons has a book that everybody would like to read and then to discuss it with some of the others.

Write a program that:

1. Finds out a way of handing around the book so that everyone gets it only once and passes it to a friend of his, and it returns to its owner at last.

2. Divides the persons into S subgroups for discussing the book. Everyone must have no more than P enemies in the subgroup he joins.

It is supposed that S*P >= K.

PROBLEM 4.

Let's consider messages written by using only the capital letters /A-Z/ and the eight symbols . , + - : / ? !

These messages are sent through a communication channel as a sequence of bytes. The number of 1's in each byte must be even.

1.Propose a coding for the above characters by binary sequences, such that unambiguous decoding is assured and the least possible number of bits is sent through the channel.

2. Write a program that:

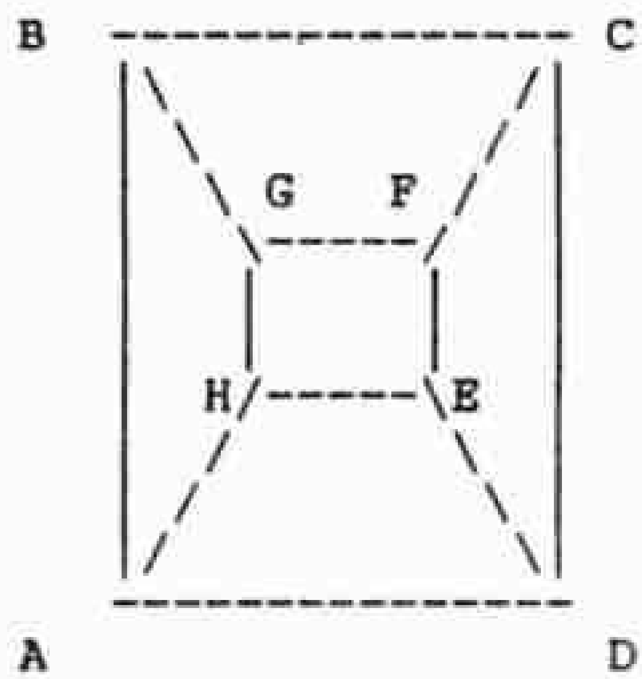
    2.1. Given the text of a message prints its encoded form ready for sending as a sequence of hexadecimal digits.

    2.2. Given a received encoded message decodes it and displays the original text.

PROBLEM 5.

Let's consider a plane graph with n vertices, each of which is of degree 3.

Example:



Let the vertices X,Y and Z be adjacent to the vertex T. We say Y is the left-hand and Z the right-hand neighbour of T with respect to X, if the oriented angle XTZ is smaller than the angle XTY (positive being the counter-clockwise direction).

For example, E is the right-hand and G the left-hand neighbour of H in respect of A because the oriented angle AHE is smaller than the angle AHG.

Write a program that:

1. Inputs the coordinates of the graph vertices and the edges and draws it on the computer display using appropriate scale. (Edges should be displayed as straight lines.)

2. Given a pair of vertices X0 and X1 and a sequence of the letters L and R, it should find a path X0X1X2...Xn on the graph, such that:

- X0 and X1 are the first two vertices;
- Xi+1 is the left-hand or the right-hand neighbour of Xi with respect to Xi-1, depending on the (i-2)-th letter in the control sequence being L and R.

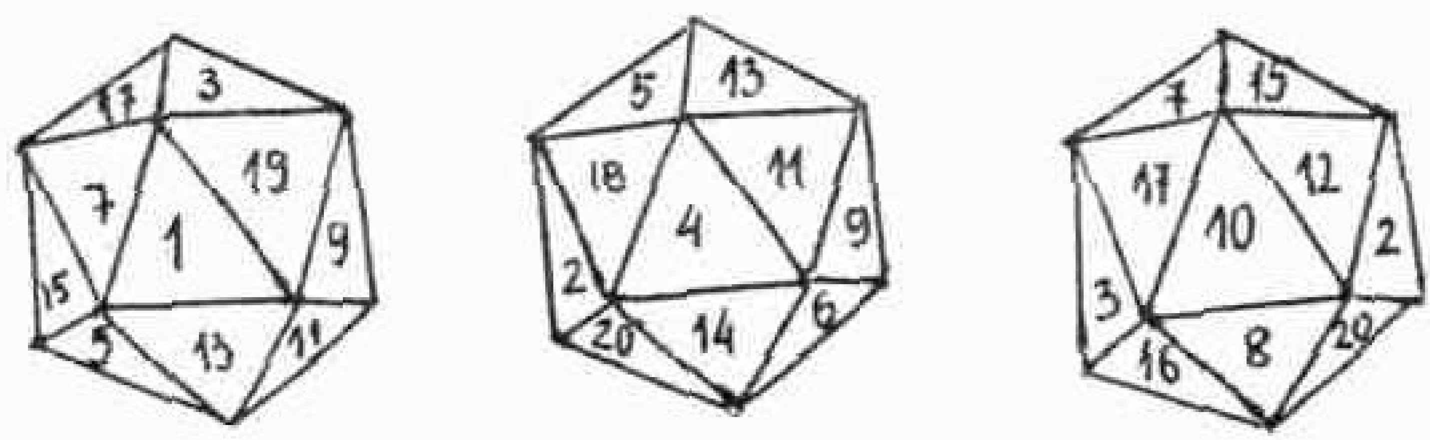
Example: The path generated for the graph from the former example, using A and H as starting vertices and the sequence LRRLLR is AHGFEDCB.

3. Draws the path found in subproblem 2 on the screen.

4. Uses a starting and an ending vertex, builds a path that goes through the least possible number of vertices, draws it on the screen and outputs the two starting vertices and the control sequence that would generate the same path as defined in subproblem 2.

PROBLEM 6.

An icosahedron is given. It is a regular polyhedron. Its sides are numbered from 1 throught 20.



The icosahedron should be routed so that to reach each side only once.

The route cost **C** is defined by the scalar product:

$$C = \sum_{i=1}^{20} i * f_i$$

where f_i is the number of the side which is reached in the i-th step.

One may pass from one side to another only if these sides are adjacent.

A. Two sides will be adjacent if there exists a common edge;

B. Two sides will be adjacent if there exists a common edge or a common point.

Find the routes with minimal costs for the cases given above.

Remark:

If for time or space complexity of the algorithm you may not find the exact solution you could propose a satisfactory one.

5. A GLIMPSE AT THE WORK OF THE JURY

The main task of the International Jury was to select the problem to be solved at the IOI'89. Six problems (Section 4) had been proposed to the attention of the Jury. Each problem had been previously examined by a member of the Coordinating Commission in order to estimate its complexity and difficulties and to analyze the possible solutions.

First the problems were reported by the respective members of the Coordinating Commission, then debates began which resulted in clarifying some important requirements concerning the problem to be chosen, namely:

- it should be clearly formulated and understandable for the participants;

- it should consist of several subproblems being of increasing complexity, so that each participant could start solving it and present a complete solution or a partial one depending on the level of his/her training.

The proposed problems were discussed as regards to their advantages and disadvantages. It was observed that Problem 4 belonged to the field of the Theory of coding, being absolutely unfamiliar to most students. Also it was feared that Problem 5 might turn out to be rather difficult for the participants for its long and complicated formulation and its troublesome mathematical base (Graph theory). Preference weighed to Problem 2, but it was seen at once that a considerable amount of input data would be needed to check the correct behaviour of the program in the great number of cases supposed to be simulated.

After the discussion a multi-round open voting was held to select the problem. Every Jury member could vote for or against each of the problems considered. After each round the problem with the greatest number of No's had to be eliminated. The order of eliminating the problems was as follows: *Problem 4*, *Problem 5*, *Problem 6*, *Problem 3*, Problem 2.

Having chosen the problem, the Jury had to formulate it clearly. A detailed discussion followed as regards to whether the value of N should be limited or whether the expression "model the exchanging" should be explained more precisely. Decisions were made again by open voting. *(The final text of the problem might be found in Section 6; it and the* original version might be compared. This comparison might reveal the *results of the* debates).

Several testing examples and an evaluating scheme (Section 8) were produced. The Jury decided that the maximal point count should be given to a participant who would implement an algorithm finding a *minimal plan without saving and displaying all the solutions.*

Next the Jury discussed in details what the procedure of accepting the students' solutions should be after the expiring of the fixed time.

According to the regulations, the participants could ask the Jury some questions in half an hour after the Olympiad had begun. Some of the questions referred to how "modelling" should be understood in this case. The reply given by the Jury read "No answer" but the Jury's members realized at once that it should have been better to stick to the clearer original explanation.

6. THE PROBLEM SELECTED FOR IOI

Given 2*N boxes in line, side by side; two adjacent boxes are empty, and the other boxes contain N-1 symbols "A" and N-1 symbols "B".

Example for N = 5.

```
-----------------------------------------
| A | B | B | A |   |   | A | B | A | B |
-----------------------------------------
```

Exchanging rule:

The contents of any two adjacent non-empty boxes can be moved into the two empty ones, preserving their order.

Aim:

Obtain a configuration where all A's are placed to the left of all B's, no matter where the empty boxes are.

Problem:

Write a program that:

1. Inputs from the keyboard the initial state as a sequence of A's and B's and zeros (for the empty boxes), and models the exchanging.

2. For a given initial state finds at least one exchanging plan, which reaches the aim or reports that such a plan does not exist. The output should contain the initial state, the intermediate states after each step, and the final state.

3. Finds a plan reaching the aim with a minimal number of steps.

Results:

Present at least one solution for the example mentioned above.

7. SOME SOLUTIONS OF THE IOI PROBLEM

First the solution of E.Todorov (BG) will be considered. It is the most complete solution and it is the only one that includes an analysis of the solved problem thus demonstrating the existence of a solution for n>3.

The case when n=1 is not considered. For *n=2 there are two* alternatives: either the symbols are ordered or they are not ordered. In *the second case the problem does not permit any solution.*

Special attention should be paid to the case when n=3. Then, there exist sequences which cannot be ordered and sequences that can be ordered.

Examples: 1) OOABAB cannot be ordered.
2) BBOOAA can be ordered.

Next it will be pointed out that there exists a solution for n>3. Suppose that each of the k boxes contains the symbol "A" (k>=O). It will be shown how to put the next symbol "A" *(if any) in the* (k+1)-box.

The following sequence is obtained:

$$\underbrace{\text{AAAA...AA}}_{k}\text{XXX...X} \qquad k>=0, \quad X \in \{A, B, O\}$$

If the (k+1) box contains the symbol "A" the problem is solved. In the other cases the following subcases exist:

1) The (k+1) symbol is "O";

2) The (k+1) symbol is "B", followed by the symbol "O";

3) The (k+1) symbol is 'B', followed by a symbol, which differs from the symbol "O".

Next each of the above mentioned cases will be considered.

Case 1.

```
AAA...AAOOXXX...X
---- ---
    k
```

The symbols "X" are scanned in order to find the first pair of successive symbols, where "A" is the first one. Next the pair thus found is put in the boxes having (k+1,k+2) positions respectively. If such a pair does not exist, then:

1.1. k=n−1 – the sequence is ordered.

1.2. k=n−2 – the last ordered symbol takes the last position, i.e. the following sequence:

```
AAA...AOOBBB...BA
---- --  --- ---
  n-2      n-1
```

is obtained. Next:

```
AAA...ABABB...BOO
---- --
  n-2

AAA...ABOOBB...BAB
---- --    --- --
  n-2        n-3

AAA...ABBBOOB...BAB
---- --
  n-2

AAA...AOOBBB...BAB
---- --
  n-2

AAA...ABBB...BOO
 ---- ------ --
   n-1     n-1
```

Thus the case 1.2 is completed.

Case 2.

```
AAA...ABOOXX...X
---- --
    k

AAA...ABXXOOXX...X
---- --
    k

AAA...AOOXBXX...X
---- --
    k
```

which in fact is the Case 1 considered above.

Case 3.

```
AAA...ABYXXX...X          k>=0
---- --                   Y ∈ {A,B}
    k                     X ∈ {A,B,O}
```

The positions of the pair "BY" and the first pair, beginning with the symbol "A" are changed. This case is also reduced to Case 1 considered above. It is obvious, that C * n moves (C is a constant and it does not depend on n) are needed in order to obtain the desired sequence of symbols.

Thus the analysis of problem 2 is completed.

The number of changes needed to order the given sequence by using the algorithm described above is denoted by **m**. In general m is not the minimal number having this property.

Let **M** *denote the minimal number of steps, needed to order the* symbols. It follows that $M <= m$. In order to find the moves, generating the sequence, all possible solutions are generated and the one consisting of m steps is considered.

In this case the "backtracking" method is applied. In order to minimize the number of retrievals, the following break condition is used:

$$p <= min\{m,t\}$$

where: t is the minimal solution found up to that moment;

m is the minimal number of steps of the current solution.

The program is given below:

```
PROGRAM PROBLEM ;

Var
   box,box1     : array [1..1000] of char ;
   st,stt       : array [1..1000] of byte ;
   spn,n,max    : integer ;
   em,em1,mm       : integer ;
   flag : boolean ;

 PROCEDURE INPUT ;
 var i : integer ;
 begin
    em := 0 ;
    clrscr ;
    write ('n = ') ;
    readln (n) ;
    writeln ;
    for i := 1 to 2*n do
       begin
          write ('box ',i,' ') ;
          readln (box[i]) ;
          box1[i] := box[i] ;
          if (box[i]='0') and (em=0) then em := i ;
       end ;
    em1 := em ;
 end ;

 FUNCTION CHECK : boolean ;
 var i : integer ;
     lst : char ;
     fl : boolean ;
 begin
    fl := true ;
    lst := box1[1] ;
    for i := 2 to 2*n do
       begin
```

```
         if (box1[i]='a') and (1st='b') then fl := false ;
         if box1[i]<>'0' then 1st := box1[i] ;
      end ;
   check := fl ;
end ;

PROCEDURE PRINT ;
var i : integer ;
begin
   for i := 1 to 2*n do
      write (box1[i],' ') ;
   writeln ;
end ;

PROCEDURE MOVE (pos:integer) ;
begin
   spn := spn + 1 ;
   box1[em1] := box1[pos] ;
   box1[em1+1] := box1[pos+1] ;
   box1[pos] := '0' ;
   box1[pos+1] := '0' ;
   em1 := pos ;
   print ;
end ;

PROCEDURE MOVE1 (pos:integer) ;
begin
   spn := spn + 1 ;
   box1[em1] := box1[pos] ;
   box1[em1+1] := box1[pos+1] ;
   box1[pos] := '0' ;
   box1[pos+1] := '0' ;
   em1 := pos ;
   st[spn] := pos ;
end ;


PROCEDURE FINDWAY ;
var k,t : integer ;
    flag : boolean ;
begin
   spn := 0 ;
```

```
   k := 0 ;
   while (not check) and (k<n-1) do
      begin
         if box1[k+1]='a' then k := k+1
            else begin
               if box1[k+1]='b' then
                  if box1[k+2]<>'0' then move (k+1)
                     else begin
                        move (em1+2) ;
                        move (k+1) ;
                     end ;
               t:=k+1 ;
               repeat
                  t := t+1 ;
               until box1[t]='a' ;
               if t < 2*n then move (t)
                  else begin
                     move (t-1) ;
                     move (k+2) ;
                     move (k+4) ;
                     move (k+1) ;
                     if (not check) then
                        move (2*n-1) ;
                  end ;
               k := k+1 ;
            end ;
      end ;
end ;

PROCEDURE BACK ;
var i,j : integer ;
begin
   em1 := em ;
   for i := 1 to 2*n do
      box1[i] := box[i] ;
   j := spn-1 ; spn := 0 ;
   if j = 0 then flag := false
      else begin
         for i := 1 to j do
            move1 (st[i]) ;
      end ;
end ;
```

```
PROCEDURE FORWRD ;
var i,t : integer ;
begin
   if spn<mm then
      begin
         t:=1 ;
         while (box1[t]='0') or (box1[t+1]='0') do
            t := t+1 ;
         move1(t) ;
      end
      else begin
        repeat
         t := st[spn] ;
         back ;
         repeat
            t := t+1 ;
         until (t=2*n) or ((box1[t]<>'0') and
(box1[t+1]<>'0')) ;
         if t<2*n then flag := true ;
        until (t<2*n) or (not flag) ;
        if t<2*n then move1(t) ;
       end ;
  end ;
```

```
PROCEDURE FINDMIN ;
var
     i : integer ;
begin
   if check then begin
    mm := spn ;
    em1 := em ;
    for i := 1 to 2*n do
       box1[i] := box[i] ;
    max := mm+1 ;
    spn := 0 ;
    flag := true ;
    while (flag) and (spn<=mm) do
```

```
      if check then
         begin
            if max>spn then
               begin
                  max := spn ;
                  for i := 1 to max do
                  stt[i] := st[i] ;
                  mm := max ;
               end ;
            forwrd ;
         end
         else forwrd ;
     writeln ;
     writeln (max) ;
     for i := 1 to 2*n do
        box1[i] := box[i] ;
     em1 := em ;
     print ;
     for i := 1 to max do
        move (stt[i]) ;
  end
  else writeln ('no way') ;
 end ;


BEGIN
   input ;
   print ;
   findway ;
   findmin ;
END.
```

The program of T. Tonchev (BG-2, first prize) solves the problem by implementing the classical BREADTH-FIRST SEARCH algorithm. The array **a** contains the vertices of the tree and the array **next** contains the parent of the corresponding vertex. The program checks the input data completely.

```
program olymp;
uses crt;
type arr = array[1..20] of -1..1;{ one sequence }
```

```
    ar   = array[0..3000] of arr;{ list of sequences}
    next  = array[0..3000] of integer;{ list of parents }
var  a : ar;
    n : integer;

  procedure input(var s : arr;var n : integer);
  { performs data input }
  var i,a,b,c : integer;
      err     : boolean;
      ch      : char;
  begin
    clrscr;
    repeat
      write('N = '); readln(n);
    until n in [2..10];
    i:=1;
    repeat
      if i=1 then
        begin
          a:=0; b:=0; c:=0;
        end;
      write(' Box ',i,' = ');
      repeat
        ch:=readkey; ch:=upcase(ch);
      until ch in ['A','B','0'];
      writeln(ch);
      case ch of
        'A' : s[i]:=-1;
        'B' : s[i]:=1;
        '0' : s[i]:=0
      end;
      case s[i] of
        -1 : begin
               a:=a+1;
               err:=(a>=n)
             end;
         1 : begin
               b:=b+1;
               err:=(b>=n)
             end;
         0 : begin
               c:=c+1;
```

```
            if c=2 then
              err:=(s[i-1]<>0)
            else
              err:=(c>2);
          end;
      end;
      if err then
        begin
          writeln('Error - too many A, B or 0');
          write(' or invalid place of 0');
          writeln(' Sorry,but you should begin');
          write(' again from box 1...');
          i:=1
        end
      else
        i:=i+1;
  until i>2*n;
end;

procedure fill(var a : ar;n:integer);
{ solves the problem }
var k,l,q,l1,b,j : integer;
    nx           : next;

  procedure scr(l : integer);
  { output one sequence }
  var i : integer;
  begin
    write('|');
    for i:=1 to 2*n do
      case a[l][i] of
         0 : write(' 0 |');
        -1 : write(' A |');
         1 : write(' B |')
      end;
    writeln
  end;

  function check(l : integer):boolean;
  { checks whether the sequence is ready }
  var s,b : integer;
  begin
```

```
      s:=0; b:=1;
      while a[l][b]<>1 do
        begin
          s:=s-a[l][b];
          b:=b+1
        end;
      check:=(s = n-1)
    end;

    function eq(l : integer):boolean;
    { checks if the l-th sequence is in the list }
    var i : integer;

      function comp(i,j : integer):boolean;
      { compares two sequences }
      var k : integer;
      begin
        comp:=true;
        for k:=1 to 2*n do
          if a[i][k]<>a[j][k] then
            begin
              comp:=false;
              exit
            end;
      end;

    begin
      eq:=true;
      for i:=l-1 downto 0 do
        if comp(i,l) then
          begin
            eq:=false;
            exit
          end;
    end;

  begin
    k:=0;
    l:=1;
    q:=0;
    if not check(0) then
    repeat
```

```
  b:=1;
  while a[k][b] <> 0 do
{ finds the position of the empty boxes }
    b:=b+1;
  for j:=1 to 2*n-1 do
    if abs(b-j)>1 then
  {   generates all possible sequences,}
  { starting from the current (k-th) }
     begin
        a[l]:=a[k];
        a[l][b]:=a[l][j];
        a[l][b+1]:=a[l][j+1];
        a[l][j]:=0;
        a[l][j+1]:=0;
        if check(l) then q:=l;
        if eq(l) then
          begin
            nx[l]:=k;
            l:=l+1;
          end;
      end;
  k:=k+1;
until (k>=l) or (q>0);
if k>=l then
  writeln('It''s impossible!')
else
  begin
    for l:=1 to n*8+1 do
      write('-');
    writeln;
    scr(q);
    for l:=1 to n*8+1 do
      write('-');
    writeln;
    while q>0 do
    begin
      for l:=1 to n*8+1 do
        write('-');
      writeln;
      q:=nx[q];
      scr(q);
      for l:=1 to n*8+1 do
```

```
          write('-');
        writeln;
      end;
    end;
  end;

begin
  input(a[0],n);
  fill(a,n);
  repeat until keypressed;
end.
```

The winners of the other three first prizes, namely: A. Cepaitis (USSR I), M. Kuhn (FRG) and I. Maly (CSSR) solved the problem by implementing the BACKTRACK algorithm. M. Kuhn's well written and commented program is given below.

```
PROGRAM solution_of_IOI_89;

(*      Author: Markus Kuhn, Federal Republic of Germany
   Team leader: Dr. Peter Heyderhoff                        *)

CONST n2max = 20;      (* The maximum of 2*n *)
      gap   = 0;       (* The number of moves without
improvement
                          allowed between improvements *)

TYPE box_sequence = ARRAY[1..n2max] OF CHAR;
     moves        = ARRAY[1..100] OF INTEGER;

VAR A                   : box_sequence;
    M                   : moves;
    best_solution       : moves;
    length_of_best_sol  : INTEGER;
    n                   : INTEGER;
    i                   : INTEGER;
    best                : INTEGER;
    dummy               : BOOLEAN;

PROCEDURE Print_sequence(S : box_sequence);
```

```
  VAR i : INTEGER;

  BEGIN
  FOR i:=1 TO 2*n DO
    WRITE(S[i]);
  WRITELN
  END;


FUNCTION value(S : box_sequence) : INTEGER;

  VAR i : INTEGER;  (* the start of a series of equal boxes
*)
      j : INTEGER;  (* the length of this series *)
      v : INTEGER;  (* the temporary value *)
      c : CHAR;

  BEGIN
  i:=1;
  v:=0;
  WHILE i<=2*n DO
    BEGIN
    c:=S[i];
    j:=1;
    WHILE (i+j<=n*2) AND (S[i+j]=c) DO
      j:=j+1;
    IF c<>'0' THEN v:=v+SQR(j);
    i:=i+j;
    END;
  value:=v;
  END;


PROCEDURE Move(VAR S : box_sequence;
               VAR possible : BOOLEAN; left : INTEGER);

  (* The contents of the boxes left and left+1 are taken
     in the empty boxes. If this is not possible, the
     variable possible ist set to false.                 *)

  VAR left_free : INTEGER;
```

```
  BEGIN
  possible:=(S[left]<>'0') AND (S[left+1]<>'0');
  IF possible THEN
    BEGIN
    left_free:=1;
    WHILE S[left_free]<>'0' DO
      left_free:=left_free+1;
    S[left_free  ]:=S[left  ];
    S[left_free+1]:=S[left+1];
    S[left  ]:='0';
    S[left+1]:='0'
    END;
  END;


FUNCTION test(S : box_sequence) : BOOLEAN;

  (* is S a solution ? *)

  VAR i,j : INTEGER;


  BEGIN
  test:=FALSE;
  i:=1;
  IF S[i]='0' THEN i:=3;
  IF S[i]='A' THEN
    BEGIN
    j:=0;
    WHILE (S[i]='A') OR (S[i]='0') DO
      BEGIN
      IF S[i]='A' THEN j:=j+1;
      i:=i+1;
      END;
    test:=j=n-1
    END
  END;


PROCEDURE Try(level : INTEGER; not_improved : INTEGER);

  (* This is the backtracking procedure *)
```

```
VAR  B          : box_sequence;
     i          : INTEGER;
     possible   : BOOLEAN;
     val        : INTEGER;
     flag       : BOOLEAN;
     temp       : INTEGER;

BEGIN
B:=A;
FOR i:=1 TO 2*n-1 DO
   BEGIN
   Move(A,possible,i);
   IF possible THEN
      BEGIN
      M[level]:=i;
      IF test(A) THEN
         BEGIN
         IF level<length_of_best_sol THEN
            BEGIN
            length_of_best_sol:=level;
            best_solution:=M
            END
         END
      ELSE
         BEGIN
         val:=value(A);
         flag:=best<val;
         IF (not_improved<gap) OR flag THEN
            BEGIN
            IF flag THEN
               BEGIN
               temp:=best;
               best:=val
               END;
            IF flag THEN
               Try(level+1,0)
            ELSE
               Try(level+1,not_improved+1);
            IF flag THEN best:=temp;
            END;
         A:=B
```

```
        END
      END
    END
  END;

BEGIN
WRITELN('IOI 1989');
WRITELN('--------');
WRITELN;
REPEAT
  WRITE('n = ');
  READLN(n);
  IF n<0 THEN WRITELN('The number must not be negative !');
  IF n*2>n2max THEN WRITELN('Sorry, that is too much for me
!');
UNTIL (n>=0) AND (n*2<=n2max);
WRITELN;
WRITELN('Please enter the contents of the following
boxes:');
FOR i:=1 TO 2*n DO
  REPEAT
    WRITE(i:3,' : ');
    READLN(A[i]);
    A[i]:=UPCASE(a[i]);
  UNTIL A[i] IN ['A','B','0'];
WRITELN;

best:=value(A);
length_of_best_sol:=MAXINT;
Try(1,0);

WRITE('Initial state: ');
Print_sequence(A);
WRITELN;
IF length_of_best_sol=MAXINT THEN
  WRITELN('There is no solution')
ELSE
  BEGIN
  WRITFLN('The following exchanges are necessary:');
  WRITELN;
  FOR i:=1 TO length_of_best_sol DO
```

```
    BEGIN
    Move(A,dummy,best_solution[i]);
    WRITE(i:3,' : box ',best_solution[i]:2,' and ',
          best_solution[i]+1:2,'   => ');
    Print_sequence(A);
    END
  END;
WRITELN;
END.
```

8. TEST EXAMPLES, EVALUATING PRINCIPLES AND RESULTS

Four test examples were prepared by the Jury so that to check the program behavior in various cases. Each test example determined the value of N and the initial state as a sequence of A's, B's and zeros for the empty boxes.

TEST EXAMPLE 1.

N=5 0 0 A B A B A B A B

The solution had to be obtained in 4 steps.

TEST EXAMPLE 2.

N=5 A B B A 0 0 A B A B

The minimal number of steps had to be 3.

TEST EXAMPLE 3.

N=3 0 0 A B A B

A message for no existence of a solution was expected.

TEST EXAMPLE 4.

N=4 0 A B A 0 B A B

A message for incorrect input data was expected.

The Jury decided the maximum number of points to be 100, which should be distributed as follows:

Subproblem 1. Up to 10 points.

Subproblem 2. Up to 40 points:

- up to 15 points for finding at least one plan or up to 20 points for all the plans found out;

- up to 20 points for reporting the lack of solution;

Subproblem 3.

- up to 15 points for an attempt made for optimization;

- up to 40 points for complete optimization.

Other 10 points were planned to be given in addition – 5 points if some results had been obtained after executing the program and 5 points for good programming style, an original solution, etc. (at decision of the Jury).

International Olympiad in Informatics
May 16 - May 20, 1989, Pravetz, Bulgaria

General Personal Rank List

N	Per. Id	Family Name	Country Code	P 1	P 2	P 3	Total Score	Place	Medal
1	54	Tonchev	BG2	10	40	50	100	1	G
2	4	Szabo	H	10	40	45	95	2	G
3	6	Maly	CS	5	40	50	95	2	G
4	34	Kuhn	D	10	40	45	95	2	G
5	46	Cepaitis	SU	5	40	50	95	2	G
6	48	Todorov	BG	5	40	50	95	2	G
7	49	Altanov	BG	5	35	50	90	3	S
8	50	Marinov	BG	10	40	40	90	3	S
9	23	Schwetlick	DDR	10	35	38	83	4	S
10	25	Nielaender	DDR	8	38	35	81	5	S
11	45	Novick	SU	8	30	42	80	6	S
12	27	Zhai	PRC	10	30	35	75	7	B
13	26	Yang	PRC	10	40	24	74	8	B
14	28	Zhuang	PRC	8	40	24	72	9	B
15	11	Lerios	GR	10	35	20	65	10	B
16	20	Nguyen Linh	VN	10	35	20	65	10	B
17	33	Kuehn	D	10	30	25	65	10	B
18	67	Ludvig	C	10	25	25	60	11	B
19	7	Solc	CS	10	38	10	58	12	
20	8	Kasal	CS	8	28	20	56	13	
21	53	Bonev	BG2	8	28	20	56	13	
22	32	Woehlbier	D	10	20	25	55	14	
23	3	Ladanyi	H	8	25	21	54	15	
24	12	Mavroidis	GR	10	15	25	50	16	
25	16	Stencel	PL	10	20	15	45	17	
26	66	Arguelles	C	5	20	20	45	17	
27	24	Balfanz	DDR	10	25	8	43	18	
28	72	Kolybin	SU3	5	15	20	40	19	
29	15	Kaczmarski	PL	8	25	5	38	20	
30	52	Petrov	BG2	9	18	5	32	21	
31	41	Ivanov	SU2	8	15	7	30	22	
32	68	Vidal	C	10	15	5	30	22	
33	40	Garanin	SU2	5	15	7	27	23	
34	19	Nguyen Nam	VN	5	15	0	20	24	
35	70	Pejchinov	YU	10	5	5	20	24	
36	42	Storozhuk	SU2	5	5	5	15	25	
37	44	Dogolacky	SU	5	5	5	15	25	
38	13	Stavropoulos	GR	0	0	10	10	26	
39	21	Nguyen Duc	VN	5	5	0	10	26	
40	69	Petrovski	YU	5	5	0	10	26	
41	38	Mtigwe	ZIM	7	0	0	7	27	
42	17	Lasota	PL	0	5	0	5	28	
43	36	Majoni	ZIM	5	0	0	5	28	
44	37	Lalloo	ZIM	5	0	0	5	28	
45	71	Andreev	SU3	5	0	0	5	28	
46	73	Sannikov	SU3	0	5	0	5	28	

International Olympiad in Informatics
May 16 - May 20, 1989, Pravetz, Bulgaria

General Country Rank List

N	Country Name /Leaders	Family Name	Id	Score	Place	Medal
1	Bulgaria Team Leader: Azalov P.	Todorov E. Altanov A. Marinov I.	48 49 50	95 90 90	2 3 3	G S S
		Team:	BG	275	1	*
2	China Team Leader: Wu W. Deputy Leader: Ling Q.	Yang H. Zhai H. Zhuang J.	26 27 28	74 75 72	8 7 9	B B B
		Team:	PRC	221	2	*
3	Fed.Rep. of Germany Team Leader: Heyderhoff P.	Woehlbier J. Kuehn U. Kuhn M.	32 33 34	55 65 95	14 10 2	 B G
		Team:	D	215	3	*
4	Czechoslovakia Team Leader: Demacek O.	Maly I. Solc V. Kasal S.	6 7 8	95 58 56	2 12 13	G
		Team:	CS	209	4	*
5	German Democr. Rep. Team Leader: Fothe M.	Schwetlick H. Balfanz D. Nielaender U.	23 24 25	83 43 81	4 18 5	S S
		Team:	DDR	207	5	*
6	Soviet Union Team Leader: Kirjuchin V.	Dogolacky I. Novick L. Cepaitis A.	44 45 46	15 80 95	25 6 2	 S G
		Team:	SU	190	6	*
7	Bulgaria II Team Leader: Manev K.	Petrov T. Bonev S. Tonchev T.	52 53 54	32 56 100	21 13 1	 G
		Team:	BG2	188	7	*
8	Hungary Team Leader: Toeroek T. Deputy Leader: Zsako L.	Ladanyi J. Szabo D.	3 4	54 95	15 2	 G
		Team:	H	149	8	*

International Olympiad in Informatics
May 16 - May 20, 1989, Pravetz, Bulgaria

General Country Rank List

N	Country Name /Leaders	Family Name	Id	Score	Place	Medal
9	Cuba Team Leader: Bittencourt D.	Arguelles F. Ludvig O. Vidal R.	66 67 68	45 60 30	17 11 22	B
		Team:	C	135	9	*
10	Greece Team Leader: Kilias C. Deputy Leader: Georgopoulou A.	Lerios A. Mavroidis D. Stavropoulos M.	11 12 13	65 50 10	10 16 26	B
		Team:	GR	125	10	*
11	Vietnam Team Leader: Dam H.	Nguyen Nam T. Nguyen Linh A. Nguyen Duc H.	19 20 21	20 65 10	24 10 26	B
		Team:	VN	95	11	*
12	Poland Team Leader: Waligorski S.	Kaczmarski A. Stencel K. Lasota S.	15 16 17	38 45 5	20 17 28	
		Team:	PL	88	12	*
13	USSR,Pereslavl Zal. Team Leader: Zaidelman Y.	Garanin I. Ivanov E. Storozhuk I.	40 41 42	27 30 15	23 22 25	
		Team:	SU2	72	13	*
14	USSR, Paris Embassy Team Leader: Abdrachmanov A.	Andreev A. Kolybin A. Sannikov A.	71 72 73	5 40 5	28 19 28	
		Team:	SU3	50	14	*
15	Yugoslavia Team Leader: Petrushev M.	Petrovski M. Pejchinov Z.	69 70	10 20	26 24	
		Team:	YU	30	15	*
16	Zimbabwe Team Leader: Mahere S.	Majoni S. Lalloo A. Mtigwe C.	36 37 38	5 5 7	28 28 27	
		Team:	ZIM	17	16	*

Date: 19.05.1989 Time: 13:39:01

9. IOI ORGANIZATIONAL STRUCTURE

Organizing Committee

D.Donchev - Chairman

A. Boichev	P. Martinov
R.Rusev	P. Kenderov
M.Draganov	p. Azalov
B.Stamov	G. Lambreva
P.Batanova	O. Daskarev
E.Stoyanova	S. Nikolova

Executive Bodies

Chairman of the Jury

Prof. Petar Kenderov

Deputy Chairman of the Jury

Nelly Maneva

Scientific Committee

A.Goshev

E.Kelevedjiev

F.Zlatareva

I. Derzhansky

I. Nenova

K.Kolchev

K. Manev

N.Maneva - Chairman

P. Azalov

P.Kopanov

Z.Vassilev

10. WORKING GROUP FOR COORDINATING THE ACTIVITIES REFERRING TO ORGANIZING FUTURE IOI

PROF. KENDEROV PETAR
BULGARIA - 1113 SOFIA
ACAD. G.BONCHEV STR., BL.8,
INSTITUTE OF MATHEMATICS,
BULGARIAN ACAD. OF SCIENCES,
TEL.: 70-10-72
TELEX 22628 BANMAT BG;
FAX (359-2) 75-20-78

MR. POKROVSKY ALEXANDRE
UNESCO - ED/STE UNESCO
7, PI. DE FONTENOY,
75 700 PARIS, FRANCE
TEL.: (33) 1 45 68 08 36
TELEX: UNESCO TELEX
FAX: 33 1 45 67 89 0

MR. PITA
CUBA
MINISTERIO EDUCATION - LA HABANA
TEL.: 302833

DR. HEYDERHOFF PETER
FRG - G.M.D. POB 1240
D-5205
ST.AUGUSTIN
TELEX: 889469 GMDD
TEL.: (49) 2241142494

PROF. KILIAS CHRISTOS
GREECE
MAUROMICHALI 2
10679 ATHENS - GREECE
TEL.: 01-3645274

PROF. WALIGORSKI STANISLAW
POLAND - PL 00901
WARSAW PKIN P.850
TEL.: 268258
TELEX: 815591 INFUW PL

DR. KIRJUCHIN VLADIMIR
USSR
115409 MOSCOW
KASHIRSKOE SH.31 MIFI
TEL.: 3249115

MR MAHERE STEPHEN MAX
ZIMBABWE
MINISTRY OF EDUCATION, BOX 8335
CAUSEWAY, HARARE
TEL.: 792671/EXT.203

MR. CHEN SHUKAI
CHINA
SECRETARY GENERAL, C.C.F.,
COMPUTER FEDERATION, BEIJING, P.R.CHINA

APPENDIX 1.

OPEN COMPETITION ON PROGRAMMING

Sofia, May 17-19, 1987

BULGARIA

This competition was organized just before and in connection with the Second International Conference and Exhibition on CHILDREN IN THE INFORMATION AGE, which took place in Sofia, Bulgaria from May 19 till May 23, 1987. Participants in the competition were 28 secondary school students from 6 countries: Czechoslovakia, Federal Republic of Germany, Hungary, Romania, USSR and Bulgaria. The students were divided into three age groups: age group 0 (less than 14), age group 1 (less than 16) and age group 2 (less than 18 years). The problems for all age groups were prepared and selected by an International Jury on the basis of problems proposed by the team leaders of the participating countries. All problems were hard and connected with real-life applications. Students worked on them within four hours. Their final results could be qualified as "excellent". Some of them discovered original approaches to the complete solutions. Others had found the right way but did not have enough time to solve the problems comletely. Even the students who were not quite successful in this competition had shown by their work that they completely deserve to be participants in an International Competition on Programming.

Additional information about the competition is the following:

I. Organized by
- Ministry of Culture, Science and Education
- Union of the Mathematicians in Bulgaria
- Institute of Mathematics, BAS

II. Executive Bodies

1. President of the Open Competition on Programming
   Acad. Blagovest SENDOV

2. Vice President
   Prof Dontcho DONTCHEV
3. Chairman of the Jury
   Prof Petar KENDEROV
4. Deputy Chairman of the Jury
   Zdravko Vassilev
5. Coordinating Commission
   F. Zlatarova
   N. Maneva
   A. Goshev
   N. Koumanov
   V. Vutov
   Z. Vasilev, chairman

III. List of participants

BG1 Bulgaria 1

Age group 1
1. Atanas Georgiev
2. Boris Medarov

Age group 2
1. Vulcho Vulchev
2. Angel Georgiev

BG2 Bulgaria 2

Age group 1
1. Valentin Tanev
2. Svetlozar Nestorov

Age group 2
1. Ivan Batanov
2. Plamen Nikolaev

CZ Czechoslovakia

Age group 1
1. Novak Michal
2. Dobos Andrej

Age group 2
1. Tayari Jakub
2. Vasely Vladimir

FRG FR Germany

Age group 1
1. Gutchke Markus
2. Sperber Michael

Age group 2
1. Mueller Thomas
2. Mueller Mischa

HU1 Hungary 1

Age group 1
1. Farkas Karoly
2. Schadt Gyorgy

Age group 2
1. Kovats Laszio
2. Boros Peter

RO1 Romania 1
Age group 0
1. Petresku Razvan
2. Slatineanu Sebastian

RO2 Romania 2
Age group 0
1. Jigorea Razvan
2. Dimitrescu Razvan

SU Soviet Union
Age group 1
1. Evsjuhin Dmitrij
2. Shirouhov Andrei

Age group 2
• 1. Abramenkova Svetlana
2. Petrov Dmitrij

IV. Results

RESULTS – AGE GROUP 0

Other participants

Jigorea Razvan – Romania 2
Retresku Razvan – Romania 1
Dimitresku Razvan – Romania 2
Slatineanu Sebastian – Romania 1

RESULTS – AGE GROUP 1

First prize

Gutschke Markus – FR Germany

Second prize

Evsjuhin Dmitrij – Soviet Union
Dobos Andrei – Czechoslovakia

Third prize

Sperber Michael – FR Germany
Svetoslav Nestorov – Bulgaria 2

Other participants

Schadt Gyorgy – Hungary 1
Atanas Georgiev – Bulgaria 1
Valentin Tanev – Bulgaria 2
Farkas Karoly – Hungary 1
Boris Medarov – Bulgaria 1
Shirouhov Andrej – Soviet Union
Novak Michal – Czechoslovakia

RESULTS – AGE GROUP 2

First prize

Vuicho Vulchev – Bulgaria 1

Second prize

Mueller Thomas – FR Germany

Third prize

Vasely Vladimir – Czechoslovakia

Other participants

Ivan Batanov – Bulgaria 2

Plamen Nikolaev – Bulgaria 2
Mueller Mischa – FR Germany
Angel Georgiev – Bulgaria 1
Abramenkova Svetlana – Soviet Union
Boros Peter – Hungary 1
Kovats Laszlo – Hungary 1
Tayari Jakub – Czechoslovakia
Petrov Dmitrij – Soviet Union

V. Competition Problem

Lets the bus stops in a city be denoted by the integers from

$N = < 1, 2, \ldots, n >.$

Also, let

$M_1 = < i_{11}, i_{12}, \ldots, i_{1m1} >$

$M_2 = < i_{21}, i_{22}, \ldots, i_{2m2} >$

. . . .

$M_r = < i_{r1}, i_{r2}, \ldots, i_{rmr} >$

be all the bus routes in the city and i_{jk} in N, and for all $k <> 1$ it follows that : $i_{jk} <> i_{jl}$.

Each M_i is a sequence of all neighbouring bus stops in one direction only. Each bus visits all stops in the route in both directions.

Design a program to the following specifications :

Input the integers n, r and the routes M_1 , M_2 ,..., M_r .

A. Check whether one can get from any stop to any other stop by bus and display appropriate message.

B. Input two stop numbers i and j, and display all possiblle ways of getting from stop i to stop j by bus.

C. Given the stops i and j, find fastest possible travel route by bus from i to j. All times of travel between neightboring stops on the same line are roughly equal and 3 times less than the time to change busses.

The following is the welcome address to all participants written by Eugenia Sendova, who is well known for her great enthusiasm and appreciation of everything related to informatics and school.

Welcome, welcome to our friends
representing many trends
coming to a competition
we hope it will become tradition.
Is it not a great idea
to gather all of you in here
to show that you are very clever
and you'll become good friends forever.
Higher, stronger, further, wiser -
friendship is the best adviser.

APPENDIX 2

INTERNATIONAL COMPETITION FOR TECHNICAL SCHOOL STUDENTS

I. Organized by

- Ministry of Culture, Science and Education

II. Executive Bodies

1. Chairman of Jury
   Pavel Azalov
   Associate professor, Ph, D

2. Deputy Chairman of the Jury
   Evgeni Gentchev
   Associate professor, Ph, D

3. Members of the Jury
   Team Leaders

4. Coordinating Commission
   Antoan Goshev
   Ognian Gavrailov
   Elena Racheva
   Antoni Antonov

III. List of partisipants :

1. CUBA
Team Leaders : Emilio Del Canal Calleja
Sergio Vilgosola Martines
Students : Sergio Martines
Alexiel Matos
Alexander Peter
Manuel Iglesias

2. GERMAN DEMOCRATIC REPUBLIC
Team Leaders : Horst-Peter Kurbel
Peter Munk
Students : Jens Kevsel
Jugo Loof
Igor Nyssen

3. HUNGARY
Team Leaders : ,Laszlo Zsako
Andre Zibaren
Students : Tibor Krivan
Zsolt Biro
Gobor Karuczka

4. POLAND
Team Leaders : Andrzey Walat
Adam Jeske
Students : Marcin Wojas
Witold Kolodziejczyk
Tomasz Borkovski

5. SOVIET UNION
Team Leader : Victor Sokolov
Students : Eduard Zotov
Ruslan Kondratsky
Igor Davidovich

6. BULGARIA
Team Leader : Peter Kopanov
Students : Georghi Rivov
Anatoli Thoev
Pavlin Kostov

IV. RESULTS

1. Georghi Rivov – Bulgaria – 14p
   Marcin Wojas Poland – 14p
2. Alexiel Matos – Cuba – 12p
3. Pavlin Kostov – Bulgaria – 10p

1. Bulgaria – 27p
   Poland – 27p
2. Cuba – 21p
3. GDR – 14p
4. Hungary – 10p
5. USSR – 8p