Olympiads in Informatics, 2008, Vol. 2, 75–89

Tasks and Training the Youngest Beginners for Informatics Competitions

Emil KELEVEDJIEV
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
Akad. G. Bonchev str., block 8, 1113 Sofia, Bulgaria
e-mail: keleved@math.bas.bg

Zornitsa DZHENKOVA
Mathematical High School
2 Elin Pelin str., 5300 Gabrovo, Bulgaria
e-mail: zornica.dzhenkova@gmail.com

Abstract. Training children for participation in informatics competitions and eventually in the IOI has been moving to younger ages and now is starting in many countries at a level of about 5–6th grades (about 11–12 years old). The main tools for teaching and preparation are tasks. We present the experience and problems given in the Bulgarian national competitions in informatics for school students in the mentioned age group. Some features of the Bulgarian system for the preparation of the youngest school students are discussed. The study covers a period from 2001 up to present. In the paper, an attempt is made to arrange and classify tasks by keywords. As examples, selected task descriptions and comments are given.

Key words: tasks in competitive informatics, informatics for the youngest school students.

1. Introduction

In recent years, the competitions in informatics have been continually expanding and involving more and more younger students. This process can be observed in Bulgaria, as well in many other countries in the world. An example for this development is the establishment in 2007 at Belgrad, Serbia, of a new kind regional Balkan Youth Olympiad in Informatics for the students up to 15.5 years old. In Bulgaria after 2001, several age group systems have been applied to divide school students for the national informatics competitions (the Autumn, Winter, and Spring Tournaments, as well for the three rounds of the National Olympiad in Informatics).

In 2001, we had an age group of 5–7th school grades (11–13 years old), which we denoted at that time as a “youth age group”. Starting in 2002, groups were introduced with letter names: A, B, C, and D, which comprised 11–12, 9–10, 7–8, and 4–6th school grades, respectively (In Bulgarian schools the mentioned grades correspond to 18–19, 16–17, 14–15 and 11–13 years old students, respectively). Starting in 2004, an additional group for the youngest students was introduced, group E, comprising the 4–5th grades.

This modified the age division among groups A, B, C, and D, as 12, 11–10, 9–8 and 6–7th grades. Later, our observations showed that it would be better to change slightly this division principles and starting in the autumn of 2007, we have groups A, B, C, D, and E, that cover 11–12, 9–10, 7–8, 6, and 4–5th school grades, respectively.

A permanently open question, often asked by teachers and trainers, who are involved in the preparation of students from the youngest age group, is the question: how to choose suitable tasks? The goal is to cover such material that might be expected in real competitions. Of course, the style of the olympiads does not always allow good prediction about the task types even for the youngest students. Nevertheless, it is possible to outline some set of themes and task types, which can serve as preparation tools. One important starting point to do this selection is examining the tasks, given at the previous real competitions. Classifying them, it becomes possible to make up manuals and handbooks. In Bulgaria, recently published books (Kelevedjiev and Dzhenkova, 2004) and (Yovcheva and Ivanova, 2006) are successfully used in the preparation process for the mentioned age group including school students of about 4–6th, or even up to 7th grade.

2. Classification

After having accumulated enough tasks (Bulgarian web portal site for competitions in Informatics, 2008) previously given in competitions, it becomes possible to start an attempt for classification using keywords.

The chosen keywords indicate some basic features from 3 different points of view:

a) basic concepts of the programming language (mainly concerning C/C++ language) together with the simple data types: numbers, symbols, strings, text (as a set of strings and delimiters), one- and two-dimensional arrays, arrays of strings, and some special attention is emphasized on the sequences of input data elements;

b) basic control constructions that form a program: simple computation by a chosen formula, conditional operator (“if” operator), loop with a counter (“for” cycle), loop with a condition (“while” cycle), combination of a loop and an “if” operator, embedded loops, recursion, and reasonable use of procedures in programming (functions in C/C++ language);

c) algorithms (with respect to the involved subject): whole numbers and divisibility, digits of a number, long numbers, combinatory analysis, sorting, recursion, geometry (rectangular shapes with sides which are parallel to the coordinate axis), modeling (including date and time intervals, informative processing of texts, etc).

The choice and the amount of the keywords are not strictly determined in our next presentations. We rather assume keywords as abbreviations to point out what is the main essence of the task.

3. Exemplary Tasks

The following tasks are chosen to illustrate the use of keywords. They also present several main topics and trends in the competitive informatics for the youngest age group in Bul-

garian national competitions. At some tasks, simple input and output examples are given in order to clarify what the used keywords mean (especially for the task that require the output of a figure or digits:

3.1. *Keyword: Conditional Operator*

Task "Brick" (4–6th grades, Spring Tournament, 2002). A brick has a form of a regular parallelepiped with length x, width y, and height z. These sizes are expressed as whole numbers, less than 1000. Write a program, that inputs x, y, and z, and outputs a number, which is equal to the value of a minimal area that should be cut in sheet iron, so that the brick can be moved through the hole. While moving we assume that brick's sides remain parallel to the edges of the hole.

3.2. *Keyword: Embedded Cycles*

Task "Different ways" (4–6th grades, Winter Competition, 2002). Write a program that inputs a positive integer S, $5 \leqslant S \leqslant 50$, and outputs how many ways there are for the integer S to be presented as a sum of 3 different integers. Example input: 10, output 4. Explanation: $10 = 1 + 2 + 7 = 1 + 3 + 6 = 1 + 4 + 5 = 2 + 3 + 5$.

3.3. *Keyword: Printing out a Figure of Characters*

Task "Decreasing numbers" (4–6th grades, Round 1 of the National Olympiad, 2004). Write a program that inputs number N, $1 \leqslant N \leqslant 9$, and outputs the following figure: on the first row – all whole numbers from 1 trough N; on the second row – all whole numbers from 2 trough N; and in a similar way up to the Nth row, where should be placed the number N only.

Example input: 5
Output:
12345
2345
345
45
1

3.4. *Keyword: Dates and Hours*

Task "Airplane" (6–7th grades, Round 1 of the National Olympiad, 2007). An airplane departs at K hours and M minutes, and arrives at L hours and N minutes. Write a program that finds out how many hours and minutes the airplane has been flying, and which time (that of the departure or of the arrival) is earlier in the twenty-four-hour day period. The flight lasts less then 24 hours. Departure and arrival times are assumed to be in a same time zone. Program's input consists of four integers K, M, L, N, on a line, separated by spaces ($0 \leqslant K \leqslant 23$, $0 \leqslant M \leqslant 59$, $0 \leqslant L \leqslant 23$, $0 \leqslant N \leqslant 59$). The output has to

contain two lines. On the first line, two integers for the flight duration have to be written and they have to express hours and minutes. On the second line, one of the letters: D or A, has to be written, depending on what is earlier: departure or arrival.

3.5. *Keyword: Strings*

Task “Leftmost” (5–6th grades, Spring Tournament, 2001). Given is a string of length N, $50 \leqslant N \leqslant 255$, containing small and capital Latin letters and digits. Some characters may occur repeatedly. Write a program that inputs the string and determines which pair of equal characters is leftmost placed. That is, the found pair should have the following property: there are no identical characters placed before the first (rightmost) character of the found pair. The output should contain two integers in the range from 1 trough N, namely the positions of both found characters in the pair.

3.6. *Keyword: Texts*

Task “Words” (4–6th grades, Autumn Tournament, 2003). Write a program that inputs text of length up to 80 characters. We call a “word” a sequence of consecutive characters which does not contain spaces, and the word has to be separated by spaces from the other words. Your program has to output the same text as input but with the places of the longest and the shortest words exchanged. In case there is more than one longest and/or shortest word, the program has to exchange the last longest word with the first shortest one. If all the words have the same length, the program has to output the same text as input.

3.7. *Keyword: Modeling and Generating*

Task “One or Zero” (4–5th grades, Spring Tournament, 2006). Let us consider numbers 1, 10, 100, 1000, 10000, and so on. That is, we consider numbers, each of them starting with 1, followed by zeros. Now take number 1 and join 10 to its right-hand side, then join 100 again to the obtained new right-hand side, then join 1000, and so on, doing this many times. We can obtain a very long number: 110100100010000100000... Write a program that inputs integer N, $0 < N < 65000$, and outputs the Nth digit of the above defined long number.

3.8. *Keyword: Recursion*

Task “Sticks” (6–7th grades, Spring Tournament, 2006). We have a large enough quantity of two types of sticks – one with a length of 1 m, and the other, with a length of 2 m. The sticks of both types cannot be distinguished, except by length. Taking several sticks, we arrange them tightly in a line with a total length of N meters. In how many ways we may do this? Write a program that inputs N, $0 < N < 30$, and outputs the answer.

3.9. *Keyword: Geometry*

Task “Rectangles” (4–5th grades, Round 3 of the National Olympiad, 2006). Given are two rectangles with sizes a by b, and c by d respectively. We have to put both rectangles side by side, without overlapping, so that the obtained figure has the least possible perimeter. Write a program that inputs the values of a, b, c, and d, as whole numbers, less than 1000, and outputs the least perimeter. Example input: 5, 7, 6, 3. Output: 30.

3.10. *Keyword: Sorting*

Task “Arranging by the sum of digits” (4–6th grades, Winter Competitions, 2003). Given is an integer N, $1 < N < 20$, and a sequence of N different positive integers, whose values are less than 1000. Write a program that inputs this data and outputs the sequence with the given integers, arranged in an increasing order by the sum of their digits. If there are two integers with the same sums of the digits, the smallest integer should be placed first (to left-hand side of the biggest one). Each two subsequent integers should be separated by a space in the output.

3.11. *Keyword: Counting*

Task “Sum” (6–7th grades, Round 3 of the National Olympiad, 2006). Given are N, $1 < N < 20$, different positive integers $a_1, a_2, \ldots, a_N$, with values less than 1000. Consider all sums, in which each given integer occurs at most once. Write a program that outputs how many different values of the considered sums are possible. The program has to read by the standard input the value of N, followed by $a_1, a_2, \ldots, a_N$, all integers separated by spaces. The program has to output the result as an integer on the standard output.

3.12. *Keyword: Table*

Task “Table” (6–7th grades, Spring Tournament, 2007). Given is a table with m rows and n columns ($1 < m < 100$, $1 < n < 100$) with cells containing “0” or “1”. The cell in the upper left corner contains 1. We call two cells neighbors, if one of them is placed directly above, bellow, to the left, or to the right of the other. We call that a set of cells is contiguous, if we can start at any cell of this set and go to any other cell moving only through neighbor cells. Let us denote by S the largest contiguous set of cells containing only “1”, which includes the upper-left cell. In how many ways we can translate the set S within the table boundaries, so that each cell of S covers again a cell that contains “1”? Write a program that outputs this quantity. The program has to read by the standard input values of m and n, separated by a space and followed by m lines in the input, each containing n characters “0” or “1” without delimiters among them.

Table 1
Data types

Keyword	Number of Tasks
Numbers	62
String	27
One-dimensional array	22
Sequence	13
Characters	10
Text	9
Two-dimensional array	8
Array of strings	3
Stack	2

Table 2
Control constructions

Keyword	Number of Tasks
Loop	73
Embedded loops	35
Loop and conditional operator	18
Conditional operator	17
Function	12
Input and output files	3
Computation by formula	1

4. Study of Keywords

In (Kelevedjiev and Dzhenkova, 2008) we published a table with detailed description built on keywords for each task from the complete collection with 148 tasks, which were given at the National competitions in informatics for the age groups of 4–7th grades in Bulgaria during the period 2001–2007. The reader may refer to the English translated copy of the table in the Appendix 2. We give the cumulative data (Tables 1–3).

5. Trends

We present diagrams to illustrate observed tendencies for monotonic or periodic trends in time appearance of task types (by means of several chosen keywords) during the period 2001–2007 in the scene of the Bulgarian national competitions in informatics for the age groups of 4–7th grades (Figs. 1–6).

Table 3

Algorithms

Keyword	Number of Tasks	Keyword	Number of Tasks
Sequential processing	17	Combinatorial analysis	2
Digits from a number	16	Dynamic programming	2
Print out a figure of characters	12	Games and strategies	2
Counting	11	Geometry	2
Divisibility	10	Number systems	2
Text processing	10	Palindrome	2
Optimal elements	9	Rectangular figures	2
Logical	7	Recursion	2
Dates	6	Decomposing numbers	1
Long numbers	6	Exhaustive search	1
Sorting	4	Fractional numbers	1
Modeling	3	Parity	1
String of digits	3	Raising to a power	1

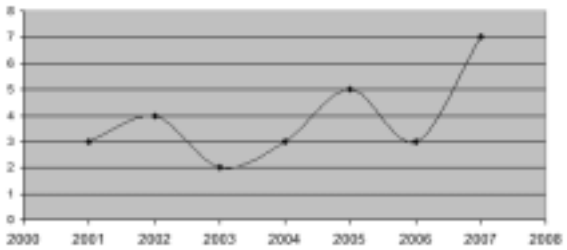

Fig. 1. Keyword: *String*.

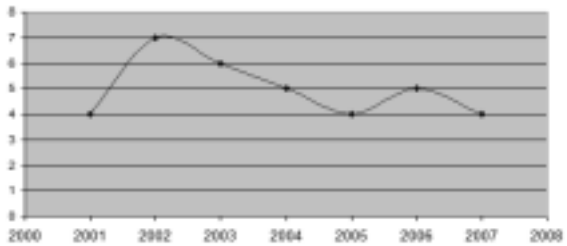

Fig. 2. Keyword: *Embedded loops*.

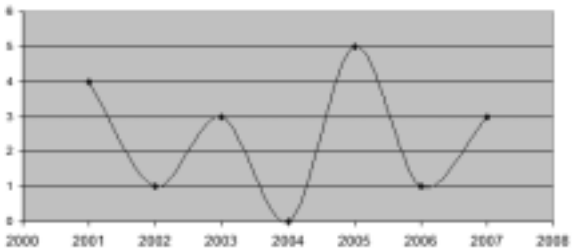

Fig. 3. Keyword: *Sequential processing*.

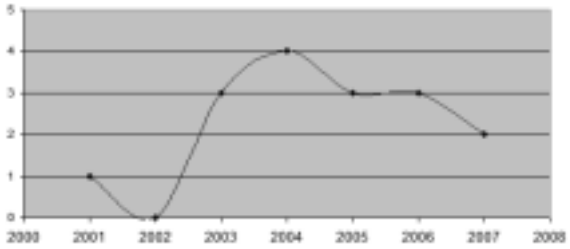

Fig. 4. Keyword: *Digits from a number*.

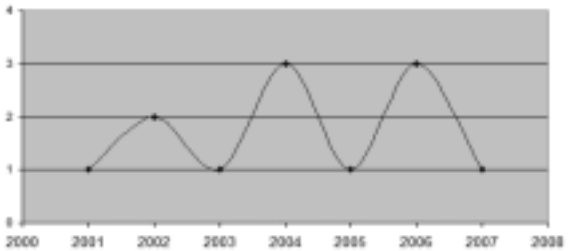

Fig. 5. Keyword: *Print out a figure of characters*.

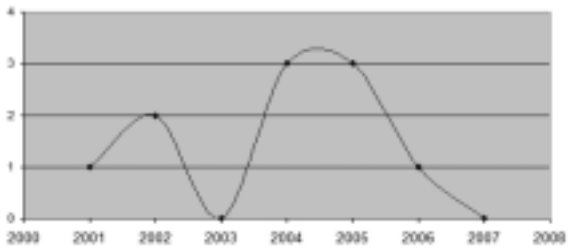

Fig. 6. Keyword: *Divisibility*.

6. Conclusions

Although the presented data as above graph samples are not statistically significant, they give us some ideas about the variety of themes.

Assigning keywords to each task is influenced by personal feelings, tastes, or opinions, but there are some more or less steady principles to choose these keywords. In many cases the keywords are self-descriptive and publishing information about tasks together with keywords is easily understandable and can help teachers in their training education process for competitive problem solving.

The authors of tasks for the Bulgarian competitions could find useful information about the history of tasks from the previous competitions in order not to duplicate or sometimes intentionally repeat some kinds of problems. In more broad sense, the study of the keywords might be applied for initializing appropriate changes and improvements in

the national curriculum which is used now as a recommendable list of themes in all the set of local out-of-class forms for young student preparation in Bulgaria. In the Appendix 1 the reader may find parts of this curriculum ((Bulgarian web portal site for competitions in Informatics, 2008; Bulgarian site for school competitions in Informatics, 2008)).

Appendix 1

Curriculum used about 2004–2005 school years:

Group E

Programming: Environment for C/C++, Branch and loop operators, Integers and Characters. One-dimensional array. Standard input and output.

Algorithms: Whole numbers arithmetic. Dates.

Geometry: Straight line coordinates.

Group D

Programming: Extended study of the programming language. Inroduction to pointers.

Data structures: Arrays and Strings. Multi-dimensional arrays. Stacks and Queues.

Methods for algorithms desing: Simple exhaustive search. Recursion. Introduction to dynamic programming. Binary search in a sorted array.

Aritmetic: Divisibility. Euclid‘s algorithm. Long integers. Number systems.

Sequences: Searching, sorting, Merging, Polynomials.

Combinatorics: Counting, Generating combinatorial configurations.

Graphs: Representations, Grid of squares.

Geometry: Coordinates in the plane. Rectangles with sides parallel to the axes.

Games: Strategies, Parity, Symmetry.

Curriculum for the National out-of-class school for preparation in informatics competitions during the 2007–2008 school years:

6th grade

	Themes	Study hours
1	Functions in C language.	2
2	One-dimensional array	2
3	Sorting	4
4	Strings	4
5	Divisibility. Prime numbers	4
6	Euclid‘s algorithm. Common Fractions	4
7	Strings in C++ style.	4
8	Two-dimensional arrays	6
9	Rectangles with sides parallel to the axes	4
10	Structures in C language	4
11	Recursion.	2
12	Number systems	6
13	Long integers	6
14	Backtracking	7
15	Grid of squares	6
	Total	65

7th grade

	Themes	Study hours
1	Parameters of the functions in C	3
2	Introduction to the standard library	2
3	Sorting – fast algorithms	2
4	Searching – binary search	2
5	Introduction to complexity of algorithms	2
6	Introduction to object-oriented programming	2
7	Combinatorial configurations	2
8	Extended Euclid‘s algorithms	3
9	Roman numerals	2
10	Polynomials	4
11	Pointers in C	2
12	Stack and Queue	3
13	Linked Lists	2
14	Searching substrings in strings	3
15	Games with numbers – using symmetry and parity	4
16	Rectangles	3
17	Bitwise operations	2
18	Long integers	3
19	Backtracking	4
20	Introduction to Dynamic Programming	5
21	Introduction to Graphs	5
	Total	60

Appendix 2

Table 4 presents all tasks given at the Bulgarian competitions during the years 2001–2007. In the column “Competition”, the names of the Autumn, Winter and Spring Competitions are abbreviated, and the three rounds of the National Olympiads in Informatics are denoted by NOI–1, NOI-2, and NOI-3, respectively.

Table 4

Tasks given at the Bulgarian competitions during the years 2001–2007

	Year	**Competition**	**Age Group**	**Task name**	**Keywords**
1	2001	Autumn	D	Stars	Characters, Embedded loops, Print out a figure of characters
2	2001	Autumn	D	Equal	Sequence, Loop and conditional operator, Sequential processing
3	2001	Autumn	D	Numbers	Numbers, Embedded loops, Digits from a number
4	2001	Winter	D	Competition	One-dimensional array, Loop, Sorting
5	2001	Winter	D	Study Circle	One-dimensional array, Loop and conditional operator

To be continued

Table 4

Tasks given at the Bulgarian competitions during the years 2001–2007 (continued)

	Year	Competition	Age Group	Task name	Keywords
6	2001	Winter	D	Text	Text, Loop, Input and output files, Text processing
7	2001	NOI-2	D	Rectangle	Numbers, Input and output files, Divisibility
8	2001	NOI-2	D	Numbers	String, Input and output files, Long numbers
9	2001	Spring	D	String	String, Embedded loops
10	2001	Spring	D	Leftmost	String, Embedded loops
11	2002	Autumn	D	Unique	String, Embedded loops, Sequential processing
12	2002	Autumn	D	Ruler	Numbers, Loop, Divisibility
13	2002	Autumn	D	Triangles	Characters, Embedded loops, Print out a figure of characters
14	2002	Winter	D	Date	Numbers, Loop, Function, Dates
15	2002	Winter	D	Largest	Numbers, Text, Embedded loops, Function, Long numbers, Combinatorial analysis
16	2002	Winter	D	Different ways	Numbers, Embedded loops, Decomposing numbers
17	2002	NOI-1	D	Longest word	Text, Loop, Function, Text processing
18	2002	NOI-1	D	Prime factors	Numbers, Embedded loops, Divisibility
19	2002	NOI-1	D	Exchanges	One-dimensional array, Loop, Combinatorial analysis
20	2002	NOI-2	D	Crossword	String, Embedded loops
21	2002	NOI-2	D	Multiplication	String, Loop, Long numbers
22	2002	NOI-2	D	Different	Array of strings, Embedded loops, Text processing
23	2002	Spring	D	Find	Numbers, Loop and conditional operator
24	2002	Spring	D	Sum	String, Loop and conditional operator, Long numbers
25	2002	Spring	D	Brick	Numbers, Logical
26	2003	Autumn	D	Words	Text, Loop and conditional operator, Text processing
27	2003	Autumn	D	Knight	Two-dimensional array, Embedded loops, Sequential processing, Geometry
28	2003	Autumn	D	Car park	Numbers, Embedded loops, Digits from a number
29	2003	Winter	D	Histogram	String, Embedded loops, Print out a figure of characters
30	2003	Winter	D	Arranged	One-dimensional array, Embedded loops, Digits from a number, Sorting
31	2003	Winter	D	Hotel	One-dimensional array, Embedded loops, Modeling
32	2003	NOI-1	D	Odd numbers	Sequence, Loop and conditional operator, Parity, Sequential processing
33	2003	NOI-2	D	Spiral	Numbers, Loop
34	2003	NOI-2	D	Trade	Numbers, Loop and conditional operator
35	2003	NOI-2	D	Cake	Numbers Embedded loops
36	2003	Spring	D	Minimax	Sequence, Loop and conditional operator, Optimal elements

To be continued

Table 4

Tasks given at the Bulgarian competitions during the years 2001–2007 (continued)

	Year	Competition	Age Group	Task name	Keywords
37	2003	Spring	D	Sum	String, Loop and conditional operator, Long numbers
38	2003	Spring	D	Street	One-dimensional array, Loop, Sequential processing
39	2004	Autumn	D	Painter	Characters, Embedded loops, Print out a figure of characters
40	2004	Autumn	D	Safe	Numbers, Loop, Digits from a number
41	2004	Autumn	E	Inequality	Numbers
42	2004	Autumn	E	Windows	Numbers
43	2004	Autumn	E	Safe	Numbers, Digits from a number
44	2004	Winter	D	Words	Text, Embedded loops, Function, Text processing
45	2004	Winter	D	Smart	Numbers, Function, Recursion
46	2004	Winter	D	Multiplication	Numbers, Loop, Divisibility
47	2004	NOI-1	D	Divisibility	Numbers, Loop, Digits from a number
48	2004	NOI-1	D	Half	String, Loop, Sequential processing
49	2004	NOI-1	D	Decreasing	Numbers, Embedded loops, Print out a figure of characters
50	2004	NOI-2	D	Game	Numbers, Loop, Divisibility
51	2004	NOI-2	D	Rooks	Two-dimensional array, Embedded loops
52	2004	NOI-2	D	Football	Numbers, Loop
53	2004	Spring	D	Fractions	Loop, Divisibility
54	2004	Spring	D	Triangles	Characters, Embedded loops, Print out a figure of characters
55	2004	Spring	D	King Artur	One-dimensional array, Loop, Digits from a number
56	2005	Autumn	D	Words	Text, Loop, Text processing
57	2005	Autumn	D	Calendar	Numbers, Embedded loops, Dates, Print out a figure of characters
58	2005	Autumn	D	Millionaire	Numbers, Loop, Dynamic programming
59	2005	Autumn	E	Bonbons	Numbers, Logical
60	2005	Autumn	E	Guess a digit	String, Loop, Function, Digits from a number
61	2005	Autumn	E	Numbers	Numbers, One-dimensional array, Loop, Digits from a number
62	2005	Winter	D	Game	String, Loop
63	2005	Winter	D	Crossword	Two-dimensional array, Embedded loops, Function
64	2005	Winter	D	Travel	Stack, Loop
65	2005	Winter	E	Windows	Numbers, Logical
66	2005	Winter	E	Minimax	Sequence, Loop and conditional operator, Optimal elements
67	2005	Winter	E	Reciprocal	Numbers, Loop, Digits from a number
68	2005	NOI-1	D	Code	Numbers, Loop, Number systems
69	2005	NOI-1	D	height	Numbers, Loop
70	2005	NOI-1	D	Triangular	One-dimensional array, Loop

To be continued

Table 4

Tasks given at the Bulgarian competitions during the years 2001–2007 (continued)

	Year	Competition	Age Group	Task name	Keywords
71	2005	NOI-1	E	Competition	Numbers
72	2005	NOI-1	E	Estimations	Sequence, Loop, Sequential processing
73	2005	NOI-1		Clock	Numbers, Divisibility
74	2005	NOI-2	D	Platforms	Two-dimensional array, Embedded loops
75	2005	NOI-2	D	Rectangle	One-dimensional array, Loop, Geometry
76	2005	NOI-2	D	Lotto	One-dimensional array, Loop
77	2005	NOI-2	E	Coating	Numbers, Divisibility
78	2005	NOI-2	E	Bus lines	Sequence, Loop, Digits from a number
79	2005	NOI-2	E	Auto	Sequence, Loop and conditional operator
80	2005	NOI-3	D	Arithmetic	Numbers, Characters, Loop
81	2005	NOI-3	D	Intervals	String, Loop
82	2005	NOI-3	D	Crossword	Array of strings, Embedded loops
83	2005	Spring	D	Game	Games and strategies, Divisibility
84	2005	Spring	D	Calendar	Numbers, Loop, Dates
85	2005	Spring	D	Monopoly	Numbers, Loop
86	2005	Spring	...	Calendar	Numbers, Dates
87	2005	Spring	E	Divisors	One-dimensional array, Loop, Divisibility
88	2005	Spring	E	Trip	Sequence, Loop and conditional operator
89	2006	Autumn	D	Library	Numbers, Loop
90	2006	Autumn	D	Trains	Numbers, Embedded loops, Print out a figure of characters
91	2006	Autumn	D	Will	Text, Loop, Text processing, Long numbers
92	2006	Autumn	E	Dates	Numbers, Dates
93	2006	Autumn	E	Text	Characters
94	2006	Autumn	E	Golden Rush	Numbers
95	2006	Winter	D	Joda	Text, Loop, Text processing
96	2006	Winter	D	Curtain	Numbers, Loop, Divisibility
97	2006	Winter	D	MAX3	One-dimensional array, Loop
98	2006	Winter	E	Animal problem	Numbers, Loop, Counting
99	2006	Winter	E	Sets	One-dimensional array, Number systems
100	2006	Winter	E	Snowflake	Characters, Embedded loops, Print out a figure of characters
101	2006	NOI-1	D	Chicken decoder	String, Loop
102	2006	NOI-1	D	meteorologist	String, Loop, Counting
103	2006	NOI-1	D	Points	Numbers, Loop, Geometry
104	2006	NOI-1	E	Arithmetic	Numbers
105	2006	NOI-1	E	Holydays	Numbers, Loop and conditional operator, Dates
106	2006	NOI-1	E	Maximal	Sequence, Loop and conditional operator, Geometry
107	2006	NOI-2	D	Diary	Numbers, Loop and conditional operator
108	2006	NOI-2	D	Roads	Numbers, Loop
109	2006	NOI-2	D	Neighbors	Two-dimensional array, Embedded loops

To be continued

Table 4

Tasks given at the Bulgarian competitions during the years 2001–2007 (continued)

	Year	Competition	Age Group	Task name	Keywords
110	2006	NOI-2	E	Square	Characters, Embedded loops, Print out a figure of characters
111	2006	NOI-2	E	Martenitza	Sequence, Loop and conditional operator, Fractional numbers
112	2006	NOI-2	E	Numbers	Sequence, Loop and conditional operator, Sequential processing
113	2006	NOI-3	D	Zig-zag	Two-dimensional array, Embedded loops
114	2006	NOI-3	D	Summer School	One-dimensional array, Loop, Sorting
115	2006	NOI-3	D	Sum	One-dimensional array, Loop, Modeling
116	2006	NOI-3	E	Cycle	Numbers Loop, Digits from a number
117	2006	NOI-3	E	Rectangles	Numbers, Geometry
118	2006	NOI-3	E	Three-digit numbers	Numbers, Loop, Digits from a number
119	2006	Spring	D	Zeros	Numbers, Loop, Raising to a power
120	2006	Spring	D	Sold	One-dimensional array, Loop
121	2006	Spring	D	Sticks	Numbers, Loop, Function, Recursion
122	2006	Spring	E	One or Zero	String, Loop, Modeling
123	2006	Spring	E	Prime factors	Numbers, Loop, Function, Counting
124	2006	Spring	E	Lucky tickets	Numbers, Loop, Digits from a number
125	2007	Winter	D	Bank Accounts	String, Loop, Digits from a number
126	2007	Winter	D	Seagull	Numbers, Characters, Loop
127	2007	Winter	D	Numbers	Loop, Sorting
128	2007	Winter	E	Text	String, Loop, Palindrome
129	2007	Winter	E	Accuracy	Numbers, Dates
130	2007	Winter	E	Ruler	Sequence, Loop and conditional operator, Geometry
131	2007	NOI-1	D	Picture	String, Embedded loops
132	2007	NOI-1	D	Teams	Numbers, Loop
133	2007	NOI-1	D	Airplane	Numbers, Loop, Dates
134	2007	NOI-1	E	Bulls	String, Digits from a number
135	2007	NOI-1	E	Coding	String, Loop
136	2007	NOI-1	E	Triangles	Numbers, Logical
137	2007	NOI-2	D	Sequence	One-dimensional array, Loop
138	2007	NOI-2	D	Group	Numbers, Loop, Function
139	2007	NOI-2	D	Paper	Text, Loop, Text processing
140	2007	NOI-2	E	Sum	Sequence, Loop, Sequential processing
141	2007	NOI-2	E	Numbers	String, Loop, Text processing
142	2007	NOI-2	E	Password	Numbers, Loop, Digits from a number
143	2007	Spring	D	Mushroom	Two-dimensional array, Embedded loops
144	2007	Spring	D	Melody	One-dimensional array, Loop
145	2007	Spring	D	Table	Two-dimensional array, Embedded loops
146	2007	Spring	E	KGB	String, Loop, Divisibility
147	2007	Spring	E	Rating	Array of strings, Embedded loops, Sorting
148	2007	Spring	E	Coloring	One-dimensional array, Loop, Sequential processing

References

Bulgarian web portal site for competitions in Informatics. Retrieved 21 April 2008 from `http://infoman.musala.com`

Bulgarian site for school competitions in Informatics. Retrieved 21 April 2008 from `http://www.math.bas.bg/infos`

Kelevedjiev, E. and Dzhenkova, Z. (2004). *Algorithms, Programs, Problems*. Manual for beginner's trainning in competitions and olympiads. Regalia, Sofia (in Bulgarian).

Kelevedjiev, E. and Dzhenkova, Z. (2008). Competition's tasks for the youngest school students. In *Mathematics, Informatics and Education in Mathematics and Informatics*. Spring Conference of the UBM, Borovetz, 2008.

Yovcheva, B. and Ivanova, I. (2006). *First Step in Programming with C/C++*. KLMN, Sofia (in Bulgarian).

E. Kelevedjiev is a research fellow in the Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences. His field of interests includes algorithms in computer science, operation research, digitization techniques, etc. He is a member of the Bulgarian National Committee for Olympiads in Informatics since 1993; leader or deputy leader of the Bulgarian teams for many IOI's and BOI's.

Z. Dzhenkova is a teacher in the Mathematical High School in Gabrovo, Bulgaria. She is coauthor of a manual for beginner's training in competitions and olympiads in informatics. Her field of scientific interests includes education in informatics and information technology; leader of school student teams and instructor in competitive informatics.