Olympiads in Informatics, 2023, Vol. 17, 131–142

DOI: 10.15388/ioi.2023.10

REPORTS

Secondary School Programming Olympiads in Gomel Region

Michael DOLINSKY,

Faculty of Mathematics and Technologies of Programming, F. Skorina Gomel State University
Sovetskaya str., 104, Gomel. 246019. Republic of Belarus
e-mail: dolinsky@gsu.by

Abstract. This article describes the content of programming competitions for students in grades 5–8 of the Gomel region. A general idea of the content of the tasks and examples of tasks by topic are given. The methodology for teaching and preparing schoolchildren for such Olympiads is also briefly described. A serious technical basis is the instrumental system of distance learning developed under the supervision of the author (`http://dl.gsu.by`).

Keywords: programming olympiads, secondary school, instrumental system of distance learning.

1. Introduction

Nowadays programming starts in elementary school (Dagienė *et al.*, 2019). It then continues into secondary school in a variety of ways: it could be machine-less learning.(Pluhar, 2021; van der Vegt, 2016), game learning (Combefis et al., 2016), using Scratch (Fagerlund *et al.*, 2020), using of specialized software development environments (Kabátová *et al.*, 2016; Tsvetkova *et al.*, 2021; Alemany *et al.*, 2016), robot programming (Kanemune *et al.*, 2017; Panskyi *et al.*, 2021).

Since September 1996, on the basis of secondary school 27 in Gomel, and in September 1999, additionally and on the basis of the distance learning site DL.GSU.BY (hereinafter referred to as DL), work is being carried out on the optional study of computer science and programming for schoolchildren of different ages (Dolinsky, 2016). The key feature of this training is the early start of education – actually from the 1st grade, and in some cases from kindergarten (Dolinsky, 2018). For such students, special programming olympiads are held in order to increase motivation for classes, as well as for the early acquisition of competitive experience. Problems for Olympiad in programming for 1–4 grades students of primary school are described at (Dolinsky, 2022). Verification of solutions is carried out automatically on the DL.GSU.BY website (Dolinsky, 2017). This article offers materials for programming Olympiads for students in grades 5–8 of secondary school and a brief description of teaching programming and preparing for such Olympiads.

Training includes a consistent study of the necessary information and fixing them by solving the proposed problems from the systematically collected problems of Olympiads of past years. Solutions are checked automatically on the DL.GSU.BY website. Note that all the Olympiad tasks of the current academic year are included in the systematic training immediately after the last Olympiad. We also note how ideas for new problems appear. On the one hand, we focus on the IOI-curriculum, and on the other hand, every year we solve the problems of the USACO, COCI, St. Petersburg individual and team Olympiads. It is they (or their subtasks) that serve as a source for the tasks of our Olympiads in subsequent years.

2. Content of the Olympiads

Tasks for grades 5–8 include 10 tasks in ascending order of complexity (each student is invited to solve all these tasks) on the following topics:

1. Introduction to programming.
2. One-dimensional array.
3. Two-dimensional array.
4. Geometry.
5. Strings.
6. Sorting.
7. Text task.
8. Elements of number theory.
9. Greedy.
10. Queue.

Topic 1: Introduction to Programming

The topic **Introduction to Programming** includes tasks in which you need to enter the initial data and output the answer, that is, to solve them, you do not need to know anything other than input and output operators. At the same time, these tasks are used for Propaedeutics of knowledge that may be needed in the future, in mathematics, programming language, Olympiad programming.

At the moment, the topic “Introduction to Programming” contains tasks on the following subtopics: formatted output; algebraic formulas; numeric operations (AND, OR, XOR, DIV, MOD, SHL,SHR); built-in functions and procedures (ABS, SQR, ODD, ROUND, TRUNC, ORD, CHR, UPCASE, STR, VAL, LENGTH, COPY, DELETE, INSERT, POS); number systems.

An example of a task on the topic “Introduction to Programming”:

Problem “Partial and Remainder”

Sample Input
7 2
Sample output:
7 div 2 = 3 7 mod 2 = 1

Sample Input:
6 4
Sample output:
6 div 4 = 1 6 mod 4 = 2

Topic 2: One-Dimensional Array

The topic **One-Dimensional Array** includes tasks to solve that require knowledge of the one-dimensional array declaration, IF condition statements, FOR and WHILE loops. At the same time, it contains tasks both for standard algorithms for processing one-dimensional arrays and tasks for Propaedeutics of useful knowledge. Currently, the topic “One-Dimensional Array” contains tasks on the following subtopics: sum; count; maximum/minimum; maximum/minimum number; cycle range; even/odd positions; long arithmetic (addition, subtraction, multiplication of a number by a digit, signs of divisibility), prefix/suffix sums, maximums, minimums; sorting by counting; cycle parameters; cyclic account; deque; search; running in a row; finding of all different numbers.

An example of a task on the topic “One-Dimensional Array”:

Problem “Divisibility by 3”

Find out if a positive integer is divisible by 3. The number can have up to 30 digits and is given by an array A of N digits. Note: A number is divisible by 3 when the sum of its digits is divisible by 3. For example, the number 159 is divisible by 3 because the sum of its digits 1 + 5 + 9 = 15 is divisible by 3. Print the remainder of the sum of its digits divided by 3 and Yes/No (divided or not given number by 3)

Input format:	**Sample Input**
N A[1] A[2] ... A[N] (numbers are entered with a space)	3 1 5 9
Output Format:	**Sample output:**
r – remainder after dividing the sum of digits by 3 Yes/No	0 Yes

Topic 3: Two-dimensional array

The topic **Two-dimensional array** includes tasks for solving which require additional knowledge of the declaration of a two-dimensional array and the use of nested loops. Currently, the topic “Two-Dimensional Array” contains tasks on the following sub-topics: sub-array – sum; sub-array – count; line count; array generation; array modification; counting along the perimeter of the array; prefix sums, maximums, minimums; strings comparison in array.

An example of a task on the topic “Two-dimensional array”:

Problem “Reset the maximum in rows”

A two-dimensional array of N*N elements is given. Zero out the first maximum element in each row.

Input Format:	**Sample Input**

N (N<=10)
a[1,1] a[1,2] ... a[1,N]
a[2,1] a[2,2] ... a[2,N]
...
a[N,1] a[N,2] ... a[N,N]
	5
	3 4 1 3 8
	2 5 6 6 5
	1 3 6 1 4
	3 5 1 7 2
	1 2 3 2 1

Output Format:	**Sample output:**

b[1,1] b[1,2] ... b[1,N]
b[2,1] b[2,2] ... b[2,N]
...
b[N,1] b[N,2] ... b[N,N]
	3 4 1 3 0
	2 5 0 6 5
	1 3 0 1 4
	3 5 1 0 2
	1 2 0 2 1

Topic 4: Geometry

The topic **Geometry** includes tasks that require the ability to solve such problems as find the distance between: two points, a point and a set of points, all points of the set, and then apply the previously studied algorithms on one-dimensional and two-dimensional arrays. In addition, this topic includes tasks on such basic concepts of geometry as perimeter and area. Currently, the topic “Geometry” contains tasks on the following subtopics: rectangle, Manhattan distance, distances from one point to set of points, the distance between the same points of two sets; neighboring distances; distances between all pairs of points; distances between all pairs of points of two sets; polygon area.

An example of a task on the topic “Geometry”:

Problem “Area under the segment”

The segment is given by the coordinates of its ends (x1, y1) and (x2, y2).

Determine the area of the figure bounded by this segment, the vertical lines from its ends and the X axis.

Notes.

1. The educated figure is a trapezoid.
2. The area of a trapezoid is half the sum of the bases times the height.

Input Format:	**Sample Input**
x1 y1 x2 y2	2 2 1 1
Output Format:	**Sample output:**
S output the answer with one decimal place	1.5

Topic 5: Strings

The topic **Strings** includes tasks for the solution of which you need to know the data types character, string, array of strings and be able to invent and debug your own algorithms. In fact, this topic is key to determining the potential abilities of the student. Currently, the topic "Strings" contains tasks on the following subtopics: cyclic shift to the right; if; string reversal; count on string; maximum per string; using of ORD; search in a string; bracket strings; lengths of array strings; count in array of strings; strings generation; strings array generation; converting a sentence into an array of words; formation of arrays of strings; subarray of characters; strings array editor; analysis of all cyclic shifts of the string.

An example of a task on the topic "Strings":

Problem "Maximum match"

N, A, B are given. A and B are strings of the same length, N is the length of these strings. Among all cyclic shifts to the right of string A, choose the one that matches the positions of the maximum number of characters with string B. Output this maximum number.

Input Format:	**Sample Input**	**Explanations**
L S1 s2	4 AGTC CTGA	
Output Format:	**Sample output:**	
Max	2	CAGT CTGA

Topic 6: Sorting

The topic **Sorting** includes tasks on the ability to apply the sorting algorithm by exchange, bubble, counting and includes tasks on the following subtopics: sorting only; sorting and output element with fixed number; fixed numbers; post-condition after sorting; sorting and output elements with variable numbers; sorting and output elements from variable range of numbers; sorting with numbers; sorting by counting; compression of coordinates; all different in ascending order.

An example of a task on the topic “Sorting”:

Problem “Compression of coordinates on a straight line”

Compress coordinates of points on a straight line. All coordinates are different integers from −10^9 to 10^9. The essence of coordinate compression is as follows: – sort all points by coordinate. instead of the coordinates of their numbers.

Input Format:	**Sample Input:**
K – number of numbers (k ≤ 10) a[1] a[2] ... a[K] – numbers separated by spaces – coordinates before compression	4 1000 -2000 -1000 2000
Output Format:	**Sample output:**
n[1] n[2] ... n[K] – coordinates after compression	3 1 2 4

Topic 7: Text problem

The topic **Text problem** contains problems with the original texts of the conditions of problems from the Belarusian Republican or Regional Olympiad (usually 1–2 pages), in which the task is changed so that its completion does not require knowledge more than that used in solving problems from topics 1–6 . The main difficulty for the participant is to single out the algorithmic statement of the problem from the textual condition. At the moment, topic 7 includes subtasks on the following algorithmic subtopics: one-dimensional array: sum, sum + condition, count, maximum, maximum number, minimum, minimum number, conditional sum, conditional minimum, selection of elements, counting sort, divisors, loops; two-dimensional array: number of maximum in a column, row sum, maximum in an array, adjacency matrix; array of strings: count by array; line count.

An example of a task on the topic “Text task”:

Task “Martian games (simplified)”

One of the most important sporting events on Mars is coming soon – the Martian Games! Athletes from all regions of Mars take part in them. Sergei Petrovich, coach of the Olymp City team, is also preparing his athletes for the Games. He trains n athletes, k of which will go to the Games. The coach faced a difficult question: it is necessary to decide who will get into the national team. Competitions are held on Mars in two stages: steeplechase and swimming, with each athlete going through both stages. The coach calculated that the i-th athlete could run in ai minutes and swim in bi minutes.

There is one feature at the Mars Games: the personal results of the athletes are practically not taken into account, and the penalty time of the team plays the main role. In competitions, the penalty time of the team is calculated according to the following formula: the product of the total running time by the total time for which all team members swam the distance. The smaller the penalty time of the team, the higher the place.

For example, let there be three athletes who ran in 3, 8, and 5 minutes, respectively, and swam in 4, 9, and 1 minute. Then the team penalty time is s = (3 + 8 + 5) * (4 + 9 + 1) = 16 * 14 = 224.

Sergey Petrovich wants to choose k athletes in such a way as to minimize the team's penalty time.

Your task is much simpler: print the number of the first athlete with the minimum total running and swimming time.

Input format

The first line of the input contains two integers n and k (1 <= k <= n <= 2000) – the number of Sergey Petrovich's athletes and the required team size for the Mars Games.

Each of the next n lines contains two integers ai and bi (1 <= ai; bi <= 10^6) – the number of minutes during which the i-th athlete will be able to run and swim the distance.

Output format

Print the number of the first athlete with the minimum total running and swimming time.

Example

Input	Output

10 8
12 1
13 4
1 33
10 10
3 6
1 19
3 12
10 10
7 7
33 2
	5

Topic 8: Elements of number theory

The topic **Elements of number theory** includes tasks that require a preliminary study of the relevant theory. Currently, the topic "Elements of Number Theory" includes subtasks on the following topics: For loop; nested For loops; while loop; For + While; dividers; simplicity test by definition; sieve of Eratosthenes; number systems; bit processing; submasks.

An example of a problem on the topic "Elements of number theory":

Problem "All submasks"

You are given a positive integer i (i < 1000). It is required to display in descending order all submasks of the binary representation of this number, starting from the number itself.

Input Format:	**Sample Input:**
I	19
Output Format:	**Sample Output:**

Ib
I1 b1
i2 b2
…
Here
Ik – decimal number in three positions
bk is its bit representation
(in the same number of bits as the original number)
	19 10011
	18 10010
	17 10001
	16 10000
	3 00011
	2 00010
	1 00001

Topic 9: Greedy algorithm

The topic **Greedy algorithm** includes tasks for the solution of which it is required to presort the input data. Currently, this topic includes tasks on the following subtopics: quadratic sorting; sorting with numbers; sorting by counting; quick sort; quick sort + while; quick sorting with numbers; comparison function; sorting array from numbers 1, 2, 3; correct bracket sequence; maximum depth of bracket expression; two arrays; selection of applications; deadline and price; minimum coverage; enumeration + greedy; stack + greedy; ad hoc.

An example of a problem on the topic "Greedy Algorithm":

Problem "Applications"

Given N applications for conducting classes in a certain audience. In each application, the beginning and end of the lesson are indicated (si and fi, respectively, for the i-th application). In the case of intersection of requests, only one of them can be satisfied. Requests with numbers i and j are joint (do not intersect) if the intervals [si, fi) and [sj, fj) do not intersect (that is, fi <= sj or fj <= si). The task is to collect the maximum number of not intersected applications.

Input format:

N
S[1] F[1]

S[2] F[2]
…
S[N] F[N]
Where:
N – Number of orders.
S[i] F[i] – description of the i-th order.
Restrictions:
1 <= N <= 200,000
0 <= S[i] < F[i] <= 100,000,000
All numbers are integers.

Output Format:
Ans – the answer to the problem – the maximum number of not intersected applications

Example:

Sample Input:	Sample output:

5
1 13
6 8
24
4 5
7 10
	3

Topic 10: Queue

The topic **Queue** includes tasks for the solution of which it is necessary to know the theory on this topic. At the moment, this topic contains tasks on the following subtopics: horse; labyrinth; three-dimensional labyrinth; pieces; three-dimensional pieces; a horse with a dynamic list of moves; numerical sequences; research with help of queue; 01-BFS; queue with bit processing.

An example of a task on the topic “Queue”

Problem “Introvert”

Ruslan is an introvert. He doesn’t like to socialize, but prefers to be alone. He lives in a cubidom with many rooms: K floors, K*K rooms on each floor. He constantly has to look for the most remote room from all the neighbors and guests. You are given a map of this cubidom, count how many moves Ruslan needs to make to reach the nearest guest or neighbor.

Input format:
The first line contains the number K (1 <= K <= 100). Next is the description of K floors separated by an empty line. Each floor is a K*K matrix. Symbol ‘*’ = empty cell ‘G’ = guest ‘S’ = neighbor ‘R’ = Ruslan. Unambiguity and correctness of tests is guaranteed.

Output Format:
Print the minimum distance from Ruslan to the nearest guest or neighbor. You can move

from any room in 6 directions (left, right, forward, backward, up a floor, down a floor) if you don't go outside the house.

Example:

Sample Input:	Sample Output:
3 G** *** *** *** *** *** *** *** **R	6

Systematic and purposeful preparation of regional Olympiads is an important means of developing the Olympiad movement in the region. Regional Olympiads are held in the Gomel region five times in the academic year: in October–November, school and city grades 1–11, and in March–April, school, city and regional (zonal) for students in grades 1–9. When conducting these Olympiads, Internet technologies and the DL.GSU.BY website are used, which allows not only schoolchildren from the Gomel region, but also everyone to participate in all the Olympiads. And, it should be noted, there are dozens of such people from all regions of Belarus and Minsk.

3. Training and Motivation System

It is important to note that, despite the focus on programming, training is essentially developing in nature and therefore it is very useful both for those who later choose information technology as their professional field, and for everyone who will be engaged in at least some time. Practice also shows that training is built in a rather interesting form. All classes are conducted only on a voluntary basis during extracurricular time. Another equally important aspect is a differentiated teaching. The use of Internet technologies makes it possible to provide individual training along a personal educational trajectory. If the student is unable to solve the problem, he is consulted by other students or the teacher. Face-to-face classes are held on Wednesdays and Sundays on the basis of the computer science cabinet of secondary school 27 in Gomel. Additionally, all students can work from home, skipping tasks that they themselves cannot solve, in order to subsequently receive help on this task in a face-to-face lesson.

In addition, weekly on weekends from Friday 8.00 to Sunday 20.00, one of the regional Olympiads that took place earlier in 2010–2022 opens for solving, solving which (at a convenient time for himself) each student can check how well he knows the material he has covered, and also what topics are still to be studied. The teacher receives similar information about each of his students.

We also note the presence of seasonal Cups (who solves more problems in learning for a certain period – autumn, winter, spring, summer from all students in grades 5–8). The awards ceremony for the top three students takes place on the first Sunday of the next season. Additionally, on the first Sunday of September, the "Person of the Year" is awarded – a student who solves the most tasks in learning for the entire academic year (autumn, winter, spring, summer). Prizes and diplomas are sent to non-resident winners by mail. These awards are provided by OpenMyGame (`http://OpenMyGame.com`), founded by graduates of our classes.

In addition, every season of the year, the student who has made the most progress during that period of time is awarded. This award is held by the leader of the circles for programming based on the use of DL in St. Petersburg (`http://vk.com/spb_dl`).

4. Conclusion

This article presents the materials of programming olympiads for students in grades 5–8 and briefly presents the methodology for teaching and preparing these students for such Olympiads. In April 2023, 3 secondary school students Gennady Martsinkevich (grade 6), Kirill Kardash (grade 7), Mikhail Brel (grade 8) became diploma winners of the Belarusian Republican Olympiad. And Mikhail Brel also got to the selection for IOI 2023. This became possible because they are ahead of the curve and participated in 9–11 grade Olympiads, which additionally contain the following topics: recursion; recurrent relations and dynamic programming; graphs; complex data structures; complex dynamic programming. In the future, we plan to include these topics in the Olympiads for grades 5–8.

References

Alemany, F.J., Vilahur, V.J. (2016). eSeeCode: Creating a computer language from teaching experiences. *Olympiads in Informatics*, 10, 3–18.

Combéfis, S., Beresnevičius, G., Dagienė V. (2016). Learning programming through games and contests: Overview, characterization and discussion. *Olympiads in Informatics*, 10, 39–60.

Dolinsky, M. (2013). An approach to teach introductory-level computer programming. *Olympiads in Informatics*, 7, 14–22.

Dolinsky, M. (2014). Technology for the development of thinking of preschool children and primary school children. *Olympiads in Informatics*, 8, 63–68.

Dolinsky, M. (2016). Gomel training school for Olympiads in Informatics. *Olympiads in Informatics,* 10, 237–247.

Dolinsky, M. (2017). A new generation distance learning system for programming and Olympiads in Informatics. *Olympiads in Informatics*, 11, 29–39.

Dolinsky, M., Dolinskaya, M. (2018). How to start teaching programming at Primary School. *Olympiads in Informatics*, 12, 13–24.

Dolinsky, M., Dolinskaya, M. (2019). Training in writing the simplest programs from early ages, *Olympiads in Informatics*, 13, 21–30.

Dolinsky, M., Dolinskaya, M. (2020). The Technology of Differentiated Instruction in Text Programming in Elementary School Based on the Website dl.gsu.by, *Olympiads in Informatics*, 14, 37–46.

Dolinsky, M.(2022). Primary School Programming Olympiads in Gomel Region (Belarus). *Olympiads in Informatics*, 16, 107–123.

Fagerlund, J., Hakkinen, P., Vesisenano, M., Viiri, J. (2020). Assessing 4th Grade Students' Computational Thinking through Scratch Programming Projects. *Informatics in Education*, 19(4), 611–640. DOI: 10.15388/infedu.2020.27

Kabátová, M., Kalaš, I., Tomcsányiová, M. (2016). Programming in Slovak Primary Schools. *Olympiads in Informatics*, 10, 125–159.

Kanemune, S., Shirai, S., Tani, S. (2017). Informatics and Programming Education at Primary and Secondary Schools in Japan. *Olympiads in Informatics*, 11, 143–150.

Panskyi, T., Rowinska, Z. (2021). A Holistic Digital Game-Based Learning Approach to Out-of-School Primary Programming Education. *Informatics in Education*, 20(2), 255–276. DOI: 10.15388/infedu.2021.12

Pluhar, Z. (2021). Extending computational thinking activities. *Olympiads in Informatics*, 15, 83–89.

Pozdniakov, S., Dagienė, V. (eds.) (2019). Informatics in schools. New ideas in school informatics. *ISSEP 2019. Lecture Notes in Computer Science*, vol 11913. Springer, Cham. https://doi.org/10.1007/978-3-030-33759-9_7

Tsvetkova, M.S., Kiryukhin V.M. (2021). Algorithmic thinking and new digital literacy. *Olympiads in Informatics*, 15, 105–118.

van der Vegt, W. (2016). Bridging the gap between Bebras and Olympiad; experiences from the Netherlands. *Olympiads in Informatics,* 10, 223–230.

M. Dolinsky is a lecturer in Gomel State University "Fr. Skoryna" from 1993. Since 1999 he is a leading developer of the educational site of the University (`dl.gsu.by`). Since 1997 he is heading preparation of the scholars in Gomel to participate in programming contests and Olympiad in informatics. He was a deputy leader of the team of Belarus for IOI'2006, IOI'2007, IOI'2008 and IOI'2009. His PhD is devoted to the tools for digital system design. His current research is in teaching Computer Science and Mathematics from early age.