Olympiads in Informatics, 2024, Vol. 18, 155–166

DOI: 10.15388/ioi.2024.12

High 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 9–11 of the Gomel region. A general idea of the thematic content of the tasks and examples of tasks by topic are given. The methodology of 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`). The paper presents very good results of Gomel schoolchildren in International Olympiad in Informatics. The main contribution of the paper is share our successful long-term experience in preparing schoolchildren for Olympiads in informatics.

Keywords*:* programming Olympiads, high school, instrumental system of distance learning.

1. Introduction

In many cases programming starts in elementary school (Dagienė *et al.*, 2019). It then continues into high school in a variety of ways: it could be unplugged learning (Plugar, 2021; van der Vegt, 2016), game learning (Combefis *et al.*, 2016), using Scratch (Fagerlund *et al.*, 2020), the use 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 programming 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).

Problems for programming Olympiads for students in grades 5–8 of secondary school are described at (Dolinsky, 2023). Verification of solutions is carried out automatically on the DL.GSU.BY website (Dolinsky, 2017). This article offers materials for programming Olympiads and a brief description of teaching programming and preparing for such Olympiads for students in grades 9–11 of high school.

Training includes a consistent study of the necessary information and solidifying 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 (IOI Syllabus, 2023), and on the other hand, every year we solve the problems of the USACO, COCI, St. Petersburg individual and team Olympiads. These tasks (or their subtasks) that serve as a source for the tasks of our Olympiads in subsequent years. It is necessary to note the inextricable connection between regional Olympiads and training. Olympiads are held to show students and teachers what topics need to be studied in order to successfully participate in Olympiads. On the other hand, all the tasks of the Olympiads upon their completion are naturally integrated into the education system. We also note that the content of Olympiads and training is expanding in accordance with the development trends of Olympiads in the world, focusing on new theoretical topics offered at the Olympiads IOI, USACO, COCI, Russian and Belarussian Olympiads in informatics and programming. When asked why we don't use many other resources in preparation for the Olympiads, we answer this way:

1. The DL.GSU.BY website has been developing since 1999, when many of the currently existing successful resources did not even exist, and we were forced to pave "our own path." As such, the author began teaching programming to schoolchildren, as well as the development and accumulation of teaching methods in the 80s of the last century.
2. We have our own teaching methodology, and we teach text-based programming, starting from elementary school (and even from kindergarten).
3. We use ideas and even tasks from other resources, but we study them in the sequence we build for studying theoretical and practical material.
4. Our best schoolchildren actively participate in Olympiads held on third-party sites, both in real time and in the mode of virtual contests and after-Olympiads solving.

2. Content of the Olympiads

Tasks for grades 9–11 include 15 tasks in ascending order of complexity (each student is invited to solve all these tasks). The first 10 tasks (the same with the corresponding Olympiad for grades 5–8) 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.

Their purpose and content are described in more detail in (Dolinsky, 2023).

Additionally, in the Olympiads of grades 9–11, 5 more tasks are given on the following topics:

1. Recursion.
2. Dynamic programming and recurrence relations.
3. Graphs.
4. Complex data structures.
5. Complex dynamic programming.

Let us describe in more detail the topics of these tasks. In most cases the condition of the task is formulated as concisely as possible, without a legend (description of the task with many sentences, that have not importance for the task). This is done in order to determine whether the student knows the corresponding algorithm and is able to write and debug a program for it or not. For every theme 11–15 theme example of theme task is given to show difficulty level of the tasks. The list of subtopics studied in each topic directly indicates the composition and order of the theory being studied. The subtopics are arranged in order of increasing difficulty, as are the tasks in each subtopic. This is what, in the opinion of the author of the article, every participant in the Olympiads needs to know. Every year the list of subtopics and tasks in them is replenished. There is no need to prove why studying such subtopics and in this order is better than other constructions. Each reader, like the author, can have his own opinion on this matter.

Topic 11."Recursion"

Topic "Recursion" includes tasks in which a recursive call to a procedure or function will be required to solve the problem. Currently, the "Recursion" topic contains tasks for the following subtopics. Set of all subsets: derivation of one of the ways to sum M, derivation of all ways to sum M, number of ways to sum M, number of ways to sum at least M, maximum sum not greater than M, subset with maximum number of matching elements, forbidden subsets, the sum of K subsets. Combinations: quantity, output. Accommodations. Permutations. Permutations with repetitions. Bracket expressions. Gray code. Fast exponentiation. Number generation. By definition. All subsets of rows of a two-dimensional array. On a two-dimensional array. Divide and conquer. Recursion with memoization. Recursion with memoization by profile.

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

Problem “Decomposition into a sum of different”

Count the number of different representations of a given natural number N as a sum of at least two pairwise distinct positive terms. Ptint out all of these.

Input Format

number N ($1 \leq N \leq 10$)

Output Format

all possible representations of the number N and the number of such representations.
Note: the output should be carried out in descending order of the terms (see the example of input – output).

Input example:

7

Sample output:

6 + 1
5 + 2
4 + 3
4 + 2 + 1
4

Topic 12.”Dynamic Programming and Recurrence Relations”

Topic ”Dynamic Programming and Recurrence Relations” includes subtopics: one-dimensional array: all sums, maximum length subsequence, number of maximum length subsequences, maximum length subsequence in O(N*LogN), knapsack, sum of several previous elements, sum of several previous elements with recovery response, maximum of sums, splitting into subarrays, prefix sums, prefix maximums, suffix minimums, permutation-number; two-dimensional array: sum of several previous elements, maximum of several previous elements, minimum sum of several previous elements, sum of maximums of several previous elements, maximum frame, prefix sums.

An example of a task on the topic “Dynamic programming and recurrence relations”:

Problem “Fibo-other numbers”

Fibo-other numbers are built according to the formula:

$f(0) = 1;$
$f(1) = 1;$
...
$f(n) = f(n-1) + f(n-2) + g(n),$

where g(n) is the number of digits in the number n.

You need to write a program that determines the number of Fibo-other numbers in a given numerical interval.

Input format:
The input consists of two numbers A and B that define the boundaries of the segment.

Output Format:
Print one number – the number of Fibo-other numbers in the interval [A, B].
Limits: 1 <= A <= B <= 1,000,000

Input example	**Output example**
2 10	3

Topic 13. "Graphs"

Topic "Graphs" includes tasks in which it is required to perform the analysis and processing of graphs and contains the following subtopics. Vertices enumeration: vertex degree, adjacency matrix. Queue: bipartite graph check, shortest paths, articulation points. Dijkstra's and Floyd's algorithms. Recursion: path with maximum number of edges, reachability matrix, sources and sinks, vertex reachability, unreachable vertices, connectivity, connected components, strongly connected components, dominant sets, cycle search, Euler cycle, Hamilton cycle, directed graph path, topological sort, maximum matching, Kuhn's algorithm. Tree definition, number of edges per vertex, tree diameter, least common ancestor, vertex visit order by depth-first search, centroid decomposition, Huffman character coding, binary tree, quaternary tree. Minimum spanning tree: Prim's and Kruskal's algorithms. Disjoint set units. Strategic games. Hidden graphs. Euler formula. Maximum flow.

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

Problem "Diameter"

Given an undirected graph G with n vertices and n – 1 edges. This graph is connected and each edge has a non-negative integer length. Let d(x, y) be the length of the shortest path between vertices x and y in graph G. The diameter of graph G is defined as the maximum of all possible distances d(x, y), where x and y are two arbitrary vertices of graph G. Write a program that which calculates the diameter of the graph G.

Input format:

Your program must read input from standard input. The first line contains a number (0 < n < 1000). The vertices of the graph are numbered from 1 to n. Each of the following n – 1 lines describes one edge: the first two numbers are the numbers of the vertices connected by this edge, and the third number is the length of this edge. The length of any edge is an integer less than 1000.

Output Format:
The single line of the standard input must contain the diameter of the graph.

Input example	**Output example**
10 4 5 5 4 3 2 4 2 1	15

5 6 4
5 1 0
5 7 4
3 8 4
3 9 3
3 10 3

Topic 14. "Complex data structures"

Topic "Complex data structures" includes tasks for which solution it is required to have the corresponding theoretical knowledge on the following topics. Segment tree: no modification (max, min, sum, max sums); single assignment (changing of one element of array): sum, minimum, maximum, minimum segment where there are all numbers from 1 to K); single increment (sum); increment on segment directly, increment on segment lazy propagation (sum, minimum, maximum, number of positives, element access); assignment on a segment (sum, number of segments from units). Fenwick tree. Trie. Bit trie. Search tree. Suffix array.

An example of a task on the topic "Complex data structures":

Problem "Tree of maximums"

Implement a data structure that stores information about N ($1 \le N \le 100000$) integers A1, ..., AN. The structure must support the following operations.

(1) INIT(N) Initialization with the number N. In this case, each number Ai is assigned the value 0.
(2) MODIFY(L, R, value) For each i, $L \le i \le R$, change Ai to Ai + value.
(3) FINDMAX(L, R) Output to the output file the maximum max{A[l], A[l + 1], ..., A[r]}.

Input Format

The input file contains no more than 100,000 operations. Each operation is described on a separate line. The operation description starts with an integer from 1 to 3 – it is number from the list above. The operation parameters follow in the order they are listed in parentheses. The numbers on each line are separated by spaces.

All operations are correct. It means that:

- The INIT operation is the very first operation in the input file and does not appear anywhere else in it.
- For the MODIFY operation, the constraints $1 \le L \le R \le N$ and $-10000 \le$ value ≤ 10000.
- The FINDMAX operation satisfies the constraints $1 \le L \le R \le N$.

Output Format

Perform the operations in the order they are listed in the input file. If you need to output some information to the output file to perform the operation, then output this information. Write the output for each operation on a separate line.

Input example	Output example
15	0
2 1 1 –6	0
3 2 4	0
3 1 2	0
3 1 3	–4
3 1 5	–4
2 2 5 –4	–8
2 4 5 –4	–8
3 1 2	
3 2 5	
2 1 3 –4	
3 4 5	
3 5 5	
2 1 1 –10	
2 1 3 3	

Topic 15. «Complex dynamic programming»

Topic «Complex dynamic programming» includes tasks that require knowledge of the relevant theory and the ability to come up with a way to apply it to the following subtopics: Bitmask DP, DP on tree, Binary Lifting, DP on profile, DP on strings, DP on numbers, DP on bit numbers, the include-exclude method.

An example of a task on the topic "Complex dynamic programming":

Problem "KT-number 2"

Numbers K and T are given. You need to find out how many T-digit numbers exist that do not contain K successive odd digits (these are the numbers 1, 3, 5, 7, 9).

Input format:

K, T, M – three numbers separated by a space.

Restrictions:

1 <= K; T <= 30; 10 <= M <= 1000000.

Output Format:

The number of T-digit numbers that do not contain K consecutive odd digits, taken modulo M (the remainder of dividing the desired number by M is implied).

Input example:	Output example:
2 3 100	50

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 a 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 approach. 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.

In addition, weekly from Thursday 8.00 to Wednesday 20.00, one of the regional Olympiads that took place earlier in 2010–2023 opens for solution, solving which (at a convenient time for himself) each student can check how well he knows the material he has studied, as well as what other topics to be studied. The teacher receives similar information about each of his students.

All of the above-described educational and competitive work is carried out on the distance learning website (Dolinsky, 2016). There, access to theory is provided, assignments are given; solutions are checked. All tests except the last one are given for educational purposes; all kinds of tables of results and ratings are built; using the forum, interactive interactions between students and each other and the teacher are supported.

To motivate the best students to become more active, they are encouraged to participate in competitions on the platform (Codeforces, 2010–2023), where they can compete with thousands of schoolchildren from all over the world. In addition, the Codeforces platform automatically recalculates the ratings of all competition participants.

In order for schoolchildren of the Gomel region to understand their level of preparation relative to their opponents, we have developed and continue to develop a website (Ratings at Codeforces of Gomel region schoolchildren, 2023). The best 25 students from Gomel are nominated to participate in the regional Olympiad. The best 25 students of the Gomel region are competing to win diplomas at the regional Olympiad. The best 15 in the Gomel region are competing to be included in the Gomel region team for the Belarusian Republican Olympiad in Informatics, the best of 11 of them are to win diplomas of the Republican Olympiad. Since 2018, schoolchildren in the Gomel region have taken 11 or more diplomas from the Republican Olympiad for 15 participants (Results of diplomas of Belarus regions, 1997–2021).

In order to make it more convenient for teachers and students to track the results of competitive and educational work of schoolchildren in the Gomel region on Codeforces (and in the future on other similar sites), we began to develop a website (Results of testing at external resources, 2023).

4. Results On International Competitions

Such a good system of training and motivation bears fruit. For the period from 1997 (when the Gomel schoolchild Kuzniatsou Artsiom first entered the IOI and won a silver medal there) to 2021 (in 2022 and 2023 Belarusian schoolchildren competed without indicating the country they represent), Belarusian schoolchildren won 88 medals, 34 of which were won by Gomel schoolchildren , the results for the regions of Belarus are presented in Table 1. (Results of Belarus regions in IOI, 1997–2021).

Twenty Gomel region schoolchildren win the medals of IOI during 1997–2021 (Results of private achievements of Gomel region schoolchildren, 1997–2021).

Gomel schoolboy, 8th grade student Mikhail Brel won a bronze medal at IOI 2023 (Results of Mikhail Brel at IOI, 2023).

The Gomel region team won a silver medal (absolute 6th place) in the twenty-fourth open All-Russian team Olympiad for schoolchildren in programming (Results of the twenty-fourth open All-Russian team Olympiad for schoolchildren in programming, 2023).

In 2016 and 2018, the author of the article became a diploma winner of the All-Russian competitions of scientific and practical works on methods of teaching computer science and informatization of education INFO-2016 and INFO-2018, held by the All-Russian Scientific and Methodological Society of Teachers and the Education and Informatics publishing house (Results XV All-Russian competition of scientific and practical works INFO (2016, 2018)).

Table 1

Results of Belarus regions in IOI (1997–2021)

Region	Total	Gold	Silver	Bronze
Gomel region	34	9	15	10
Minsk region	13	1	6	6
Vitebsk region	13	0	6	7
Lyceum BSU	12	1	4	7
Brest region	6	1	0	5
Minsk (capital of Belarus)	4	0	3	1
Grodno region	4	0	2	2
Mogilev region	2	1	1	0
Total	**88**	**13**	**37**	**38**

The respected reviewer requested a comparison of Belarus' results with other countries at the IOI. I consider this to be unlawful; the author does not teach all schoolchildren in Belarus. But in 2006–2009, the author was the coach of the Belarusian national team at IOI. And during that period, the Belarusian team won 14 medals for 16 participants (6 gold, 6 silver and 2 bronze), showing the sixth result during this period after China, Poland, Russia, USA and Taiwan (Results of countries on IOI, 2006–2009).

By the way, preparation for Olympiads in computer science and programming goes very far beyond the scope of the material studied in computer science classes in Belarusian schools.

5. Conclusion

This article presents the materials of Olympiads in programming for students 9–11 grades and briefly presents the methodology for teaching and preparing these students for such Olympiads. The Olympiad includes 15 problems on the topics: introduction to programming, one-dimensional array, two-dimensional array, geometry, strings, sorting, text problem, elements of number theory, greedy algorithm, queue, recursion, dynamic programming and recurrence relations, graphs, complex data structures, dynamic programming is hard. This approach allows us to provide the paradigm "each student will solve at least one problem", "no student will solve all problems", although in practice there are cases of violation of both rules. But in general, it turns out a balanced Olympiad for students with any level of training. The annual transfer of tasks from past Olympiads to the course "Olympiads 9–11" provides an opportunity for a systematic study of theory in preparation for subsequent Olympiads. And the weekly training Olympiads (based on the materials of the Olympiads of previous years) provide training in practical skills for solving problems at the olympiad and quality control of assimilation of the material. The paper also presented successes of Gomel region schoolchildren at international contests in informatics and programming.

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

Codeforces (2010–2023). `https://codeforces.com/`

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.

Dolinsky, M. (2023). Secondary School Programming Olympiads in Gomel Region (Belarus). *Olympiads in Informatics*, 17, 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., Kala , I., Tomcsányiová, M. (2016). Programming in Slovak Primary Schools. *Olympiads in Informatics*, 10, 125–159.

IOI Syllabus (2023). `https://ioinformatics.org/page/syllabus/12`

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.

Plugar, 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`

Ratings at Codeforces of Gomel region schoolchildren (2023). `https://dl.gsu.by/codeforces/`

Results of Belarus regions (1997–2021). (In Russian). `https://dl.gsu.by/olymp/result/ioi/region.asp`

Results of diplomas of Belarus regions (1997–2021). `https://dl.gsu.by/olymp/rgomel.asp`

Results of countries on IOI (2006–2009). `https://dl.gsu.by/olymp/result/ioi2006/`

Results of Mikhail Brel at IOI (2023). `https://stats.ioinformatics.org/people/7971`

Results of private achievement of Gomel region schoolchildren (1997–2021). `https://dl.gsu.by/servlet/olympResultsPersonalMedal?c.id=1&u.c=25&lng=rus&r.id=3&a.r=3`

Results of testing at external resources (2023). `https://dl.gsu.by/etr/`

Results of the twenty-fourth open All-Russian team Olympiad for schoolchildren in programming (2023). (In Russian). `https://neerc.ifmo.ru/school/archive/2023-2024/ru-olymp-team-russia-2023-standings.html`

Results of XV All-Russian competition of scientific and practical works INFO-2018 (2018). (In Russian). `https://infojournal.ru/competition/info-2018-result/`

Results of XV All-Russian competition of scientific and practical works INFO-2016 (2016). (In Russian). `https://infojournal.ru/competition/info-2016-result/`

Tsvetkova, MS, Kiryukhin VM (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.