



Zhytomyr Ivan Franko State University Journal.  
Pedagogical Sciences. Vol. 1 (108)

Вісник Житомирського державного  
університету імені Івана Франка.  
Педагогічні науки. Вип. 1 (108)

ISSN (Print): 2663-6387  
ISSN (Online): 2664-0155

**UDC 004.588**

**DOI 10.35433/pedagogy.1(108).2022.46-60**

## **FEATURES OF THE FOUR-LEVEL APPROACH TO TRAINING FOR STUDENT OLYMPIADS IN COMPUTER SCIENCE**

**S. D. Vapnichnyi\***

*The article proposes elements of the methodological system of training students for computer science Olympiads (CSO), which covers four levels: basic, medium, high and advanced. For each of the levels, a list of topics is given; for some topics there are proposed assessment tasks, that can be used to determine the student's readiness. Also for each level the sets of tasks on eolymp.com and codeforces.com are offered. To increase the effectiveness of the proposed approach, it is recommended that students, starting from the medium level, participate in various competitions and Olympiads, which give many motivating impulses to master programming. High and advanced levels include own topics and tasks as well.*

*Besides, the paper formulates the requirements that were the basis for the selection of tasks by topics and separation of the proposed levels. In particular, there are proposed the following requirements for tasks: availability of interesting plot, making curiosity for students, illustrativeness, connectivity and diversity. The construction of task systems takes into account the didactic and methodological requirements that determine the pedagogical expediency of their use. The model of the process of creating a tasks system, which consists of analytical, design and technological stages, is described.*

*The proposed approach methodology was successfully used in teaching students of Uzhhorod specialized boarding school with in-depth study of certain subjects (UzhSBSwiDSCS), at summer and winter programming schools in Kremenchug and summer programming schools in Khust. Many students of UzhSBSwiDSCS studied on the basis of this approach, participated in competitions and became winners of various stages of the All-Ukrainian Student Olympiad in Informatics. The mentioned approach has been successfully tested in distance learning.*

**Key words:** *computer science, elements of methodological system, Olympiad, programming schools.*

\* Senior Lecturer  
(Uzhhorod National University)  
svapnichnyi@gmail.com  
ORCID: 0000-0001-8131-0884

## ОСОБЛИВОСТІ ЧОТИРИРІВНЕВОГО ПІДХОДУ ПІДГОТОВКИ УЧНІВ ДО УЧАСТІ В ОЛІМПІАДАХ З ІНФОРМАТИКИ

С. Д. Вапнічний

У статті запропоновано елементи методичної системи підготовки учнів до олімпіад з інформатики, яка охоплює чотири рівні. Виокремлено такі рівні, як базовий, середній, високий та продвинутий. Для кожного з рівнів наведено перелік тем та запропоновано контрольні задачі для деяких тем, за якими можна визначити підготовленість учня. Також для кожного рівня запропоновано набори задач на двох відомих ресурсах – `euolymp.com` та `codeforces.com`. Для підвищення ефективності запропонованого у роботі підходу рекомендується учням, починаючи з середнього рівня, брати участь в різного роду змаганнях та олімпіадах, які дають багато мотивуючих поштовхів до опанування програмування. Високий та продвинутий рівні включають в себе і власні авторські теми та задачі.

У статті наведено вимоги, якими користувались, при підбірці задач по темах та виділенню рівнів. Зокрема, виділено такі, як наявність цікавої фабули та виникнення інтересу в учнів, ілюстративність, зв'язність та різноманітність. При побудові систем завдань враховуються дидактичні та методичні вимоги, що визначають педагогічну доцільність їх використання. Описано модель процесу створення системи завдань, яка складається з аналітичного, проєктувального та технологічного етапів.

Запропонований підхід успішно використовувалась при навчанні учнів Ужгородської загальноосвітньої спеціалізованої школи-інтернату з поглибленим вивченням окремих предметів (УЗСШЗПВОП), на літніх та зимових учнівських школах з програмування в Кременчуці та на літніх учнівських школах з програмування в Хусті. Багато учнів УЗСШЗПВОП навчались за запропонованим підходом, брали участь в олімпіадах і ставали призерами різних етапів Всеукраїнської учнівської олімпіади з інформатики. Також вказаний підхід пройшов успішне апробування в умовах дистанційного навчання.

**Ключові слова:** інформатика, елементи методичної системи, олімпіада, школи з програмування.

**Introduction of the issue.** The global IT industry is growing rapidly and needs more and more employees. These workers do not need to build large factories, as in the days of industrialization, but just ordinary office space and computer equipment. But the training of such a specialist is much more difficult than an employee in the industry. This training should start at school. Various competitions, tournaments and Olympiads contribute to the development of motivation to engage in programming in the initial stages.

**Current state of the issue.** There is a number of publications related to this topic. In [1] theoretical information on three main functions of teaching programming is highlighted and presented: general educational, developmental and upbringing. It is stated that the general educational function of teaching programming in

**Постановка проблеми.** Світова ІТ-галузь стрімко зростає і потребує все більшої і більшої кількості працівників. Для цих працівників не потрібно, як в часи індустріалізації, будувати великі заводи, а достатньо звичайного офісного приміщення та комп'ютерної техніки. Але підготовка такого фахівця є суттєво складнішою, ніж працівника в промисловості, ця підготовка повинна починатися ще в школі. Виробленню мотивації займатися програмуванням на початкових етапах гарно сприяють різні конкурси, турніри та олімпіади.

**Аналіз останніх досліджень і публікацій.** Існує ряд публікацій, які стосуються цієї тематики. В [1] виділено та наведено теоретичні відомості про три основні функції навчання програмування: загальноосвітню, розвивальну та виховну. Вказано, що загальноосвітня функція навчання програмування в школах полягає у формуванні в школярів знань про

schools is to form students' knowledge of the fundamental concepts and paradigms of programming and the formation of students' programming skills and abilities. The developmental function is to develop students' algorithmic thinking style, intellectual qualities and creative abilities, the formation of students' ability to see the problem at different levels of detail, the ability to use programming to solve practical problems. The upbringing function is to form in students such qualities as pedantry, discipline, accuracy, internal control, perseverance, awareness of personal responsibility for the results of their work, the desire for self-affirmation through creative activities, ability to work both individually and collectively, systematic learning.

In [2] three components of preparing students for Olympiads in informatics are highlighted: diagnostic-motivational, content-theoretical and procedural-activity (competitive-training). It is stated that diagnostic-motivational one is aimed at identifying students capable of programming and solving non-standard problems, motivating students to in-depth study of this section of computer science. Content-theoretical component consists of 2 subcomponents: learning a specific programming language and solving logical, non-standard problems using simple programming methods (array search, use of mathematical formula etc.); study of basic Olympiad algorithms and their application in solving problems. Procedural-activity (competitive-training) component is aimed at consolidating theoretical material, its application to solving Olympiad tasks of different levels of complexity, participation in Olympiads, contests, tournaments, writing computer programs. On the basis of the specified components the author's approach to training for Olympiads is offered.

In [3] the main stages of students training for participation in competitions are indicated and the classification of Internet resources that can be used in the

фундаментальні концепції і парадигми програмування та формуванні у школярів умінь та навиків програмування. Розвивальна функція полягає в розвитку у школярів алгоритмічного стилю мислення, інтелектуальних якостей та творчих здібностей, формуванню у школярів здатності бачити задачу на різних рівнях деталізації, вміння застосовувати програмування для вирішення практичних задач. Виховна функція полягає у формуванні в школярів таких якостей як педантичність, дисциплінованість, акуратність, внутрішня керованість, наполегливість, усвідомленість особистої відповідальності за результати своєї праці, прагнення до самоствердження через творчу діяльність, уміння працювати як особисто, так і в колективі, системність у навчальній діяльності.

У роботі [2] виділено три компоненти підготовки учнів до олімпіад з інформатики: діагностико-мотиваційний, змістовно-теоретичний та процесуально-діяльнісний (змагально-тренувальний). Вказано, що діагностико-мотиваційний спрямований на виявлення учнів, здібних до програмування та розв'язування нестандартних задач, мотивацію учнів до поглибленого вивчення даного розділу інформатики. Змістовно-теоретичний складається з 2 підкомпонентів: вивчення конкретної мови програмування та розв'язання логічних, нестандартних задач з використанням нескладних методів програмування (пошук у масиві, використання математичних формул тощо); вивчення базових олімпіадних алгоритмів та застосування їх при розв'язуванні задач. Процесуально-діяльнісний (змагально-тренувальний) спрямований на закріплення теоретичного матеріалу, застосування його до розв'язання олімпіадних задач різного рівня складності, участь у олімпіадах, конкурсах, турнірах, написання прикладних програм. На базі вказаних компонентів запропоновано авторський підхід підготовки до олімпіад.

У роботі [3] вказано основні етапи підготовки учнів до участі у змаганнях та розглянуто класифікацію Інтернет-ресурсів, які можна використовувати у навчальному

educational process is considered. In [4] it is described in detail the method of conducting two lessons on the topics: wave algorithm for finding the shortest path in the maze and ordering tabular quantities. In [5] a brief overview of systems of training for the Olympiads in informatics in some countries is provided. In [6] the scheme consisting of seven stages is proposed to solve the Olympiad problem. The analysis of the considered sources shows that the authors do not offer a level differentiation of methods with a description of topics and a list of corresponding tasks, that is seen as a shortcoming. Accordingly, this article proposes a methodology of training for Olympiads, taking into account the division at the levels and describes its features.

**Aim of research** is to develop elements of the methodological system, that provide a four-level approach to the training for Olympiad in informatics for secondary school students, to describe in detail the topics for each of the levels, to select tasks for each topic.

**Methodology:** analysis and generalization of experience of long-term practical training students for the Olympiads in informatics, experience of other coaches, selection and systematization of Olympiad tasks of different levels of complexity, work with scientific and scientific-methodological sources, systematic approach to teaching computer science.

**Scientific novelty:** there have been developed elements of the methodological system, that provide a four-level approach to the training for Olympiad in informatics for secondary school students; the development contains classified topics and tasks that correspond to each of the four proposed levels. These elements of methodology has been tested and proved to be quite effective in teaching students programming and training for the Olympiads in informatics.

**Results and discussion.** In training for the Olympiads in informatics, we will

процесі. В [4] детально описано методіку проведення двох уроків на теми: хвильовий алгоритм пошуку найкоротшого шляху у лабіринті та впорядкування табличних величин. У [5] наведено короткий огляд систем підготовки до олімпіад з інформатики в деяких країнах. В [6] запропоновано схему, яка складається з семи етапів, для розв'язання олімпіадної задачі. Аналіз розглянутих джерел показує, що автори не пропонують рівневої диференціації методик з описом тем та переліком відповідних задач, що вбачається недоліком. Відповідно, у цій статті пропонується методіка підготовки до олімпіад з врахуванням поділу на рівні та описуються її особливості.

**Мета статті** – розробити елементи методичної системи, що передбачають чотирирівневий підхід до підготовки до олімпіад з інформатики для учнів середніх шкіл, детально описати теми для кожного з рівнів, здійснити підбір задач з кожної теми.

**Методи дослідження:** аналіз і узагальнення досвіду багаторічної практичної підготовки учнів до олімпіад з інформатики, досвіду роботи інших тренерів, відбір і систематизація олімпіадних задач різних рівнів складності, робота з науковими та науково-методичними джерелами, системний підхід до навчання інформатики.

**Наукова новизна:** розроблено елементи методичної системи, що передбачають чотирирівневий підхід до підготовки до олімпіад з інформатики для учнів середніх шкіл; в цій розробці класифіковано теми та задачі, що відповідають кожному з чотирьох запропонованих рівнів. Ці елементи методіки пройшли апробацію і виявились досить ефективними при навчанні учнів програмування та підготовці до олімпіад з інформатики.

**Виклад основного матеріалу.** При підготовці до олімпіад з інформатики будемо виділяти такі вимоги до підбору задач за темами та визначення рівнів: 1) наявність цікавої фабули та викликання інтересу в учнів (завоювати увагу учнів і затягнути їх займатись олімпіадною інформатикою, відволікши їх від гаджетів,

highlight the following requirements for the selection of tasks by topics and determine the levels: 1) availability of interesting plot and making curiosity for students (to gain the attention of students and engage them in Olympiad informatics, distracting them from the gadgets with which they spend more and more time); 2) illustrativeness (so that it is possible to conveniently depict the task, which will allow students to better see it and possibly reduce it to an isomorphic or similar task); 3) connectivity (important topics are continued at the next level and require quality training in the previous material; for example, at the second level students study the general concepts of graphs and input / output of relevant data structures, the third level is used in the study of in-depth search, which is being used at the fourth level in finding the component of strong connectivity of the graph); 4) diversity (the system of tasks takes into account the presence of different types of thinking, types of memory etc.).

At task systems building, the requirements that determine the pedagogical expediency of their use are taken into account: didactic, reflecting the relevant traditional and specific principles of teaching, and methodological that take into account the features of informatics as a subject and science.

The model of the process of creating a system of tasks (as a set of tasks, in some way related to each other and having several levels of organization in relation to consistent subordination) consists of the following stages: 1) analytical (analysis of educational material meaning, formulation of goals and determination their mutual compliance, selection of content); 2) design (choice of methods and techniques, determination of forms of presentation of educational material, methods of its presentation); 3) technological (technical creation of task systems in accordance with the requirements).

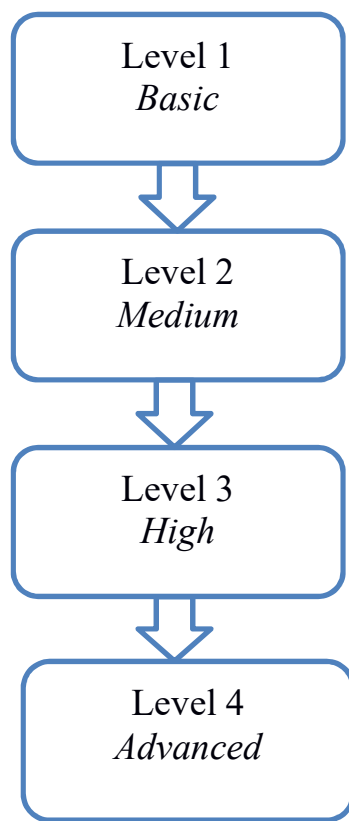
It is proposed to teach students programming according to a scheme that provides four levels: basic, medium, high and advanced.

з якими вони все більше і більше проводять час); 2) ілюстративність (щоб можна було зручно зобразити задачу, що дасть можливість краще побачити її і, можливо, звести до ізоморфної чи подібної задачі); 3) зв'язність (важливі теми мають продовження на наступному рівні і вимагають якісної підготовки з попереднього матеріалу; наприклад, на другому рівні вивчають загальні поняття про графи та ввід/вивід відповідних структур даних, на третьому це використовують при вивченні пошуку в глибину, який використовується на четвертому рівні при знаходженні компонента сильної зв'язності графа); 4) різноманітність (система завдань враховує наявність різних типів мислення, видів пам'яті і т.д.).

При побудові систем завдань враховуються вимоги, що визначають педагогічну доцільність їх використання: дидактичні, що відображають відповідні традиційні та специфічні принципи навчання, та методичні, що враховують особливості інформатики як навчального предмета та науки.

Модель процесу створення системи завдань (як сукупності завдань, певним чином пов'язаних між собою і які мають кілька рівнів організації, що знаходяться у відношенні послідовного підпорядкування) складається з наступних етапів: 1) аналітичний (аналіз змісту навчального матеріалу, формулювання цілей та встановлення їх взаємної відповідності, відбір змісту); 2) проєктувальний (вибір методів та методичних прийомів, визначення форм подання навчального матеріалу, способів його подання); 3) технологічний (технічне створення систем завдань відповідно до вимог, що пред'являються).

Пропонується навчати учнів програмування за схемою, яка передбачає чотири рівні. Це базовий, середній, високий та продвинутий.



**Fig. 1. Scheme of training on four levels**

Elements of the methodology of teaching the basics of programming, as well as literature that covers the basic level, are given in [7]. In the process of studying a certain topic, students are grouped into different age groups according to the method of Valentyn Melnyk, who has the title of People's Teacher of Ukraine [8]. The main factor in the formation of such groups is not age, but the degree of mastery of topics that corresponds to a certain level.

The basic level allows mastering the basics of programming and includes topics such as linear, conditional and cyclic constructions, working with strings and arrays, bit operations, procedures and functions. Each topic includes a number of different subtopics. In particular, the topic "Linear structures" includes the following subtopics:

- example of a simple program;
- constants and variables;
- types of variables;
- assignment operator;

Елементи методики навчання основ програмування, а також література, яка охоплює базовий рівень, наведено в [7]. У процесі вивчення певної теми учні об'єднуються у різновікові групи за методом народного вчителя України Валентина Мельника [8]. Основним чинником формування таких груп є не вікові особливості, а рівень засвоєння тем, які відповідають певному рівню.

Базовий рівень дозволяє опанувати ази програмування і включає в себе такі теми, як лінійні, умовні та циклічні конструкції, робота з рядками та з масивами, бітові операції, процедури і функції. Кожна з тем включає в себе певну кількість різних підтем. Зокрема, тема "Лінійні конструкції" включає в себе такі підтеми:

- приклад простої програми;
- константи та змінні;
- типи змінних;
- оператор присвоєння;
- арифметика у C++, арифметичні вирази та операції;
- введення, виведення у консоль у

- arithmetic in C ++, arithmetic expressions and operations;
- input, output to the console in the style of C and C ++ (printf, scanf, cin, cout threads);
- formatted input and output to the console.

The selection of tasks for these subtopics is given in Table 1.

стилі C та C++ (функції printf, scanf, потоки cin, cout);

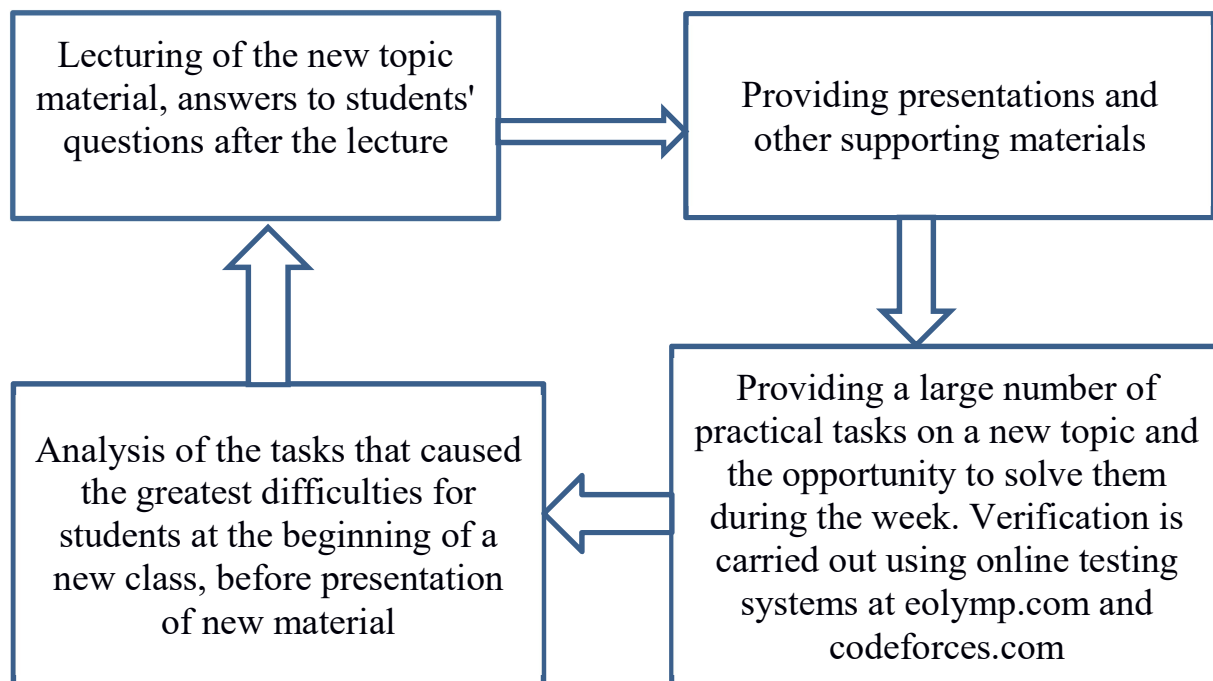
- форматоване введення та виведення у консоль.

Підбір задач до цих підтем подано в Таблиці 1.

Table 1

**Topics and tasks covered by the basic level**

№	Topic name	Task numbers on eolymp.com	Task numbers on codeforces.com
1	Linear constructions	57, 133, 157, 219, 255, 478, 933, 935, 941, 943, 945, 949, 951, 1286, 1289, 8801-8855	4A, 1335A, 1385A, 1296A, 1542A, 1506A, 1593A
2	Conditional constructions	63, 67, 108, 125, 133, 248, 653, 902, 903, 905, 915, 918, 923, 1312, 1351, 1610, 1623, 1954, 1955, 2042, 2043, 2044, 7337, 8242, 8612, 8618, 8619, 8845, 8863, 8870, 8873, 8883, 8885, 8889, 9539	50A, 282A, 1360A, 1369A, 1385A, 1186A, 119A
3	Cyclic constructions	2, 295, 388, 421, 514, 518, 520, 622, 904, 907, 914, 917, 919, 921, 928, 1118, 1603, 1605, 1607, 1609, 2370, 2607, 2863, 3133, 4101, 4721, 4751, 5283, 7365, 7829, 7843, 7844, 8243, 8533, 8544, 8545, 8546, 8630, 8631, 8681, 8682, 8897, 8900, 8917, 8926, 8941, 8946	231A, 158A, 263A, 750A, 703A, 1462A, 1382A
4	Working with strings	205, 494, 909, 1119, 2611, 7326, 8222, 8243, 8316, 8318, 8319, 8320, 8519, 8533, 8569, 8570, 8571, 8610, 8620, 8625, 8632	71A, 1607A, 1462B, 1539B, 1504A, 1301A, 1146A
5	Working with arrays	354, 8774, 7850, 3923, 2327, 2329, 986, 7829, 4730, 7831, 7832, 914, 917, 928, 1952, 7834, 7849, 5059, 2238, 7368, 7833, 922, 4760, 1460, 2098, 3935, 8548, 7537	427A, 1399A, 758A, 1367B, 1512A, 1399B, 1542A
6	Bit operations	27, 769, 1550, 1612, 1645, 1647, 1648, 1753, 2616, 2733, 2807, 4142, 5050, 5095, 5097, 5314, 5315, 5316, 5317, 5318, 5319, 5320, 5718, 5868, 6311, 6777, 7339, 9098	
7	Procedures and functions	913, 920, 926, 1209, 1648, 2862, 8239, 8240, 8241, 9026	



**Fig. 2. General scheme of work on each topic**

Each topic is processed according to the scheme shown in Figure 2.

The scheme connects the following sequence of actions that will allow effective teaching:

- Lecturing of the new topic material, answers to students' questions after the lecture.
- Providing presentations and other supporting materials.
- Analysis of the tasks that caused the greatest difficulties for students at the beginning of a new class before presentation of new material.
- Providing a large number of practical tasks on a new topic and the opportunity to solve them during the week. The easier the topic, the more tasks are provided. In particular, there may be hundreds of such tasks on the first topics of the basic level. Verification is carried out using online testing systems at eolymp.com and codeforces.com.

The task for knowledge control, which will indicate good mastery of the topic "Arrays" and a number of previous topics and, accordingly, indicate the possibility of

Кожна тема опрацьовується за схемою, наведеною на рисунку 2.

Схема пов'язує таку послідовність заходів, що дозволять здійснювати ефективно навчання:

- Виклад матеріалу нової теми, відповіді на запитання учнів після лекції.
- Надання презентації та інших допоміжних матеріалів.
- Розбір задач, що викликали в учнів найбільші труднощі на початку нового заняття, перед поданням нового матеріалу.
- Надання великої кількості практичних задач із нової теми та можливість їх розв'язувати впродовж тижня. Чим легша тема, тим більше задач надається. Зокрема, на перші теми базового рівня таких задач може бути сотні. Перевірка здійснюється за допомогою систем онлайн-тестування на eolymp.com та codeforces.com.

Задачею для контролю знань, яка вкаже на добре володіння темою "Масиви" та ряд попередніх тем і, відповідно, вкаже на можливість перейти на наступний рівень, може бути така: вивести елементи



moving to the next level, may be as follows: display elements of a given array in the following order: first minimal, then maximal, then the next minimal, then the next maximal etc.

заданого масиву у такому порядку – спочатку найменший, далі найбільший, далі наступний найменший, далі наступний найбільший і т.д.

Table 2

**Topics and tasks covered by the medium level**

№	Topic name	Task numbers on eolymp.com	Task numbers on codeforces.com
1	Recursion. Enumeration	6, 480, 1391, 1488, 1511, 1514, 1517, 2167, 25236 2764, 3603, 3606	1490D, 1167B, 1385D, 1373D, 1528A, 1461D, 768D
2	Euclidean algorithm	563, 1155, 2612	17A, 26A, 109B
3	Sieve of Eratosthenes	22, 33, 1302, 2245, 3843, 4076	237C, 615D
4	Containers STL: vector	354, 986, 2327, 2329, 3923, 7850, 8774	427A, 1399A, 758A, 1367B, 1512A, 1399B, 1542A
5	Containers STL: stack	693, 1776, 1871, 2479, 6122, 6123	5C, 343B, 281D
6	Containers STL: set	555, 790, 1225, 1226, 1227, 1228	1312B, 1277B, 975A, 960B, 978C, 977D, 975C, 982B, 966A
7	Containers STL: map	1211, 1868, 2040, 5492	4C, 855A, 903C, 918B
8	Containers STL: deque	694, 2248, 3161, 6128, 6129, 6129	1179A, 1579E1, 1579E2
9	Binary and ternary searches	312, 1506, 1516, 1522, 3326, 3966, 3967	202A, 750A, 1138A, 1476A, 1566A, 1592A, 1611B
10	Greedy algorithms. Introduction	66, 138, 182, 609, 1228, 4211, 4746	231A, 50A, 339A, 469A, 996A, 1399A, 1409A
11	Dynamic programming. Introduction	115, 263, 798, 799, 1560, 4051	996A, 1472B, 1538A, 702A, 894A, 1519B, 1462B
12	Graph theory. Introduction	292, 548, 1064, 2383, 2923, 3165, 4007	1549B, 939A, 755A, 115A, 938A, 1055A, 500A
13	Game tasks. Introduction	32, 110, 148, 308, 309, 310, 311	959A, 1480A, 1398B, 832A, 1270A, 1373B, 1419A
14	Computational geometry. Introduction	924, 925, 926, 929, 932, 934	1369A, 1398A, 1622A, 1312A, 1466A, 1422A, 1064A
15	Combinatorial tasks. Introduction	318, 1539, 5104, 5329, 5716	1499A, 1591A, 629A, 1543B, 1582B, 1166A, 1293B

The transition to the second level is also accompanied by the active participation of students in various competitions and Olympiads. Undoubtedly, the main skills that develop

Перехід на другий рівень супроводжується також активною участю учнів в різного роду змаганнях та олімпіадах. Безперечно, головні навички, які розвиває участь в олімпіадах – це власне

in competitions are programming itself and the ability to look for errors in written programs. There are different formats for holding Olympiads and evaluating solutions. But for each of them, the number of points obtained for the proposed solution directly depends on its correctness. Therefore, it is necessary to implement the idea without errors, preferably from the first time. Otherwise, the participant must quickly identify and correct inaccuracies.

The ability to debug programs quickly is one of the most important skills in programming. Undoubtedly, the winners of prestigious programming Olympiads are very talented and persistent people. There are many competitions held by the largest IT companies. Technical recruiters have been monitoring the results of various competitions and specific participants for many years. The most promising and successful are offered an internship, combining it with university studies, with the opportunity to get a full-fledged job after graduation. In general, participation in programming Olympiads consists of two stages: creating effective algorithms for tasks and their implementation.

At first glance, it may seem that to achieve significant results in the Olympiads, it is enough to study a number of existing algorithms, and then only successfully use them during the competition, leaving others no chance of winning. In fact, it doesn't work that way. Otherwise it would not be so interesting to be engaged in Olympiad informatics. Tasks are formed in such a way that it is not enough to guess the algorithm that needs to be used to solve them. Almost always for a complete solution it is necessary to upgrade a known algorithm, supplement it, combine several algorithms in one program. One can't do without inventing own new ideas.

The task for knowledge control, which will indicate good mastery of the topic "Dynamic programming" for the second level may be the following task [9, task 5101]. Hodja Nasreddin is in the upper

програмування і вміння шукати помилки в написаних програмах. Існують різні формати проведення і оцінювання розв'язків на олімпіадах. Але на кожній з них кількість отриманих балів за запропонований розв'язок напряму залежить від його правильності. Тому потрібно реалізувати ідею без помилок, бажано з першого разу. В інакшому випадку учасник повинен швидко виявити та виправити неточності. Вміння швидко налагоджувати програми – одне з найважливіших вмінь у програмуванні. Без сумніву, переможці престижних олімпіад з програмування – це дуже талановиті й наполегливі люди. Існує безліч олімпіад, які проводяться найбільшими ІТ-компаніями. Технічні рекрутери упродовж довгих років слідкують за результатами різних олімпіад і конкретних учасників. Найбільш перспективним і успішним пропонують пройти стажування, комбінуючи його з навчанням в університеті, з можливістю отримати повноцінну роботу після закінчення навчання. Взагалі, участь в олімпіадах з програмування складається з двох етапів: складання ефективних алгоритмів поставлених задач і їх реалізація. На перший погляд може здатися, що для досягнення солідних результатів на олімпіадах достатньо вивчити певну кількість існуючих алгоритмів, а потім лише успішно використати їх під час змагання, не залишаючи іншим жодного шансу на перемогу. Насправді це так не працює. Інакше займатися олімпіадною інформатикою було б не так цікаво. Задачі формуються таким чином, що недостатньо вгадати алгоритм, який потрібно використати для їх розв'язування. Практично завжди для повного розв'язку необхідно модернізувати відомий алгоритм, доповнити його, скомбінувати декілька алгоритмів в одній програмі. Без вигадування власних нових ідей не обійтися.

Задачею для контролю знань, яка вкаже на добре володіння темою "Динамічне програмування" для другого рівня може бути наступна задача [9, задача 5101]. Ходжа Насреддін знаходиться у лівій

left cell of the  $n \times n$  table, and his donkey is in the lower right. Hodja walks only to the right or down, the donkey walks only to the left or up. In how many ways can they meet in one cell? (The two methods are considered different if they have different routes of Hodge or donkey). One number  $n$  ( $1 \leq n \leq 50$ ) is given at the input. Print one number – the number of ways Hodja and Donkey will meet. Since this number can be very large, print it modulo 9929.

To solve this problem it is necessary to know the Bellman optimality principle with a fairly simple filling of a two-dimensional array.

верхній клітині таблиці розміром  $n \times n$ , а його осел – у правій нижній. Ходжа ходить лише вправо чи вниз, осел – лише вліво чи вгору. Скількома способами вони можуть зустрітися в одній клітині? (Два способи вважаються різними, якщо в них маршрути Ходжі або віслюка різні). Задано на вході одне число  $n$  ( $1 \leq n \leq 50$ ). Виведіть одне число – кількість способів, якими Ходжа та осел зустрічатимуться. Оскільки це число може бути дуже великим, виведіть його за модулем 9929.

Наведена задача потребує при розв'язуванні знання принципу оптимальності Беллмана при досить простому заповненні двовимірного масиву.

Table 3

**Topics and tasks covered by the high level**

№	Topic name	Task numbers on eolymp.com	Task numbers on codeforces.com
1	Greedy algorithms. Tasks of increased complexity	2016, 8691, 1403 4973, 7029, 7174, 7493	1632D, 1630B, 1624G, 1620D, 1620C, 1622C, 1621B
2	Dynamic programming. Quadratic and cubic varieties. Organization of recalculation by profile	764, 1553, 1283, 1559, 809, 1552, 798, 1105, 2302	1555C, 1253C
3	Graph theory. Methods for determining the shortest distances	1365 1388,1389, 2209, 2267,7710, 4856, 974, 975	1486B, 520B, 3A, 370A, 329B, 266B, 1418C
4	Graph theory. Finding the minimum skeletal tree	3385	609E, 959E,891C
5	Graph theory. Finding the smallest common ancestor	5217, 2317, 3298, 3300, 5218, 3299, 2318	1304E
6	Graph theory. Finding components of strong connectivity, bridges and connection points	674, 1943	550D, 652E, 555E, 231E
7	Graph theory. Formation of pairs	1738 1989 2904	1525D, 1630F, 739D, 86B, 1624C, 1549B
8	Algorithms for working with strings. Tasks of increased complexity	2172, 2303, 3844, 6129, 6030	126B, 1537E1, 1326D2, 1326D1, 471D, 1621I
9	Modeling. Tasks of increased complexity	3,8,35,1437,1489,1490,1493, 1496,1682,2808	1353D, 1352G, 1334C, 1348C, 1630B, 1618E
10	Automata theory.	2171	126B, 471D, 633C, 432D

	Introduction		
11	Game tasks. Tasks of increased complexity	32, 1009, 1011, 1417, 35056	268A, 455B
12	Combinatorial tasks. Tasks of increased complexity	1478, 1440, 65, 1480, 1787	1620G, 1615F, 1613F
13	Fenwick tree	3061, 3395, 4073, 5619, 6233, 309	369E, 276C
14	Segment tree	4073, 4496, 8247, 2941, 2041, 4255, 2907	339D, 356A, 459D, 61E, 380C, 474F, 292E, 501D, 220E, 338E, 19D, 351D, 515E, 540E, 609F, 594D, 455E
15	Sqrt-decomposition. Root optimization. Mo's algorithm	4255, 2961, 8275	121E, 103D, 710F
16	Isomorphic representation of problems in computer science	Own development, tasks for which are posted on <a href="http://olymp.uzhnu.edu.ua">olymp.uzhnu.edu.ua</a>	

The task for knowledge control, which will indicate good mastery of the above topic "Dynamic programming", but at the third level, may be the following task [9, task 1528]. A subsequence is formed from a string by removing zero or more characters from it. For the given three lines, you should count the number of their different non-empty common subsequences. Each test consists of three words in three different lines. The length of each word is slightly more than 50. Each word consists of lowercase Latin letters ('a' – 'z'). Print the number of different non-empty common subsequences in a separate line for each test.

To solve this problem, we need to use a three-dimensional array, the values of which will be filled using the Bellman principle of optimality.

The topic "Isomorphic representation of problems in computer science" is the author's and part of it was developed jointly with Bohdan Zadorozhny, who in 2019 won first place in the final stage of the competition for the defense of research works of the Small Academy of Sciences.

Задачею для контролю знань, яка вкаже на добре володіння вищезгаданою темою "Динамічне програмування", але вже на третьому рівні, може бути наступна задача [9, задача 1528]. Підпоследовність утворюється з рядка видаленням нуля або кількох символів із нього. За заданими трьома рядками Вам слід підрахувати кількість їхніх різних непорожніх загальних підпоследовностей. Кожен тест складається із трьох слів, які знаходяться у трьох різних рядках. Довжина кожного слова трохи більше 50. Кожне слово складається з латинських літер нижнього регістру ('a' – 'z'). Для кожного тесту вивести в окремому рядку кількість різних непорожніх загальних підпоследовностей.

Для розв'язання цієї задачі потрібно використати тривимірний масив, значення у якому будемо заповнювати, використовуючи принцип оптимальності Беллмана.

Тема "Ізоморфне представлення задач в інформатиці" є авторською і частина її була розроблена спільно з Богданом Задорожним, який у 2019 році виборов перше місце на заключному етапі конкурсу-захисту науково-дослідницьких робіт Малої академії наук.

Table 4

**Topics and tasks covered by the advanced level**

№	Topic name	Task numbers on eolymp.com	Task numbers on codeforces.com
1	Cartesian tree	1314, 5293, 686, 687, 690	431E, 420D
2	Centroid decomposition	973, 4104	293E, 321C, 322E, 342E, 914E, 960E, 990G, 1156D
3	Ad hoc – tasks	1491, 7802	1446A, 1442B, 1436D, 1438D, 1437B
4	Optimization in dynamic programming	7469	321E, 673E, 834D, 660F, 631E, 91E, 377E, 1619H
5	Automata theory. Tasks of increased complexity	2172, 2303, 3844, 6030	1537E1, 1326D2, 535D
6	Sprague-Grundi theory	2656, 2657, 5878, 5874	102059I
7	Berlekamp-Massey algorithm		506E
8	Graph theory. Flow theory	1106, 1110, 1617, 1991, 2903, 3641	1530D, 237E, 1426E
9	Graph theory. Various tasks of increased complexity	610, 1750	1406C, 1146C, 1283D, 1242B, 909E
10	Persistent data structures	1884, 2957, 4313, 2955	464E, 484E
11	Number theory. Tasks of increased complexity	8593, 1096, 1012	10C, 1389E, 1310F, 1264F
12	Approaches to reducing the running time of programs that are related to sequences and matrices	Own development, tasks for which are posted on olymp.uzhnu.edu.ua	

Note that the topic "Approaches to reducing the time of programs that are associated with sequences and matrices" is well exposed in [10]. Quite a lot of information on the topics of the fourth level is given in [11-14].

**Conclusions and research perspectives.** The proposed four-level methodological system of training students for the Olympiads in informatics was successfully used in teaching students of Uzhgorod specialized boarding school with in-depth study of certain subjects (UzhSBSwiDSCS), at summer and winter students' programming schools in Kremenchug and summer programming

Відмітимо, що тема "Підходи до зменшення часу роботи програм, які пов'язані з послідовностями та матрицями" є добре розкрита в [10]. Достатньо багато інформації за темами четвертого рівня наведено в [11-14].

**Висновки з даного дослідження і перспективи подальших розвідок.** Запропонована чотирирівнева методична система підготовки учнів до олімпіад з інформатики успішно використовувалась при навчанні учнів Ужгородської загальноосвітньої спеціалізованої школи-інтернату з поглибленим вивченням окремих предметів (УЗСШІзПВОП), на літніх та

schools in Khust. Many students of UzhSBSwiDSCS studied by this system of training, participated in Olympiads and became winners of various stages of the All-Ukrainian Student Olympiad in Informatics. This approach also has been successfully tested in distance learning [15]. Many of the students continued to participate in programming Olympiads as students of higher educational institutions. Over time, they received invitations and internships in the world's leading IT companies [16, 17].

зимових учнівських школах з програмування в Кременчуці та на літніх школах з програмування в Хусті. Багато учнів УЗСШіЗПВОП навчалися за цією системою підготовки, брали участь в олімпіадах і ставали призерами різних етапів Всеукраїнської учнівської олімпіади з інформатики. Також вказаний підхід пройшов успішне апробування в умовах дистанційного навчання [15]. Багато хто з учнів продовжував займатись олімпіадами з програмування і вже будучи студентами. З часом вони отримували запрошення та проходили стажування в провідних ІТ-компаніях світу [16, 17].

#### REFERENCES (TRANSLATED & TRANSLITERATED)

1. Gryshko, L.V. (2009). *Metodychna systema navchannya osnov programuvannya majbutnih inzheneriv-programistiv [Methodical system of teaching the basics of programming to future software engineers]*. Candidate's thesis. Cherkasy, 281 [in Ukrainian].
2. Zhukovskyj, S.S. (2013). *Pedagogichni umovy pidgotovky obdarovanyh shkolyariv do olimpiad z informatyky [Pedagogical conditions for the preparation of gifted students for computer science competitions]*. Candidate's thesis. Zhytomyr, 235 [in Ukrainian].
3. Postova, S. (2009). *Pidgotovka uchniv do uchasti v olimpiadax z informatyky ta informacijnyh texnologij z vykorystannyam internet-resursiv [Preparing students to participate in competitions in computer science and information technology using Internet resources]*. *Naukovi zapysky. Seriya: Problemy metodyky fizyko-matematychnoyi i tehnologichnoyi osvity – Academic Commentaries. Series: Problems of methodology physical-mathematical and technological education*. Kirovograd, № 8 (2), 32-38 [in Ukrainian].
4. Kaglyuk, V.V. *Metodyka pidgotovky do olimpiady z informatyky (indyvidualnyj tvorchyj projekt) [Methods of preparation for the Olympiad in Informatics (individual creative project)]*. Retrieved from: <http://oblosvita.com/navigaciya/skrynka/informatyka/11232-metodika-pdgotovki-do-olmpadi-z-nformatiki.html> [in Ukrainian].
5. Melnyk, V.I., Goroshko Yu.V., & Mitsa, O.V. (2017). *Oglyad system pidgotovky do olimpiad z informatyky v deyakyh krayinax [Review of systems for preparing for computer science competitions in some countries]*. *Suchasni informacijni texnologiji v osviti i nauci – Modern information technologies in education and science: II Vseukr. nauk.-prakt. konf. z mizhnar. uchastyu*. Zhytomyr, 21-23 [in Ukrainian].
6. Horoshko, Yu.V., Mitsa, O.V., & Melnyk, V.I. (2019). *Applying of the general scheme to solving the olympiad task on computer science*. *Information Technologies and Learning Tools*, 71.3, 40-52 [in English].
7. Vapnichnyj, S.D. (2021). *Navchannya pochatkiv programuvannya na urokah informatyky [Learning the basics of programming in computer science lessons]*. *Visnyk Nacionalnogo universytetu "Cherniqivskij kolegium" imeni T.G. Shevchenka – Bulletin of the T.H. Shevchenko National University "Chernihiv Colehium"*, № 4, 128-136 [in Ukrainian].
8. Melnyk, V.I. (2006). *Systema roboty v riznovikovyh grupah pid chas pidgotovky do olimpiad [The system of work in different age groups in preparation for Olympiads]*. *Kompyuter v shkoli ta simyi – Computer at school and family*, № 6, 25-30 [in Ukrainian].

9. *Internet portal of organizational and methodological support of remote programming competitions for gifted youth of educational institutions of Ukraine e-olimp*. Retrieved from: <https://www.e-olymp.com/> [in English].

10. Mitsa, O., Horoshko, Y., & Vapnichnyi, S. (2020). Reduction of programs execution time for tasks related to sequences or matrices. *EDP Sciences. In SHS Web of Conferences*, vol. 75, 04019 [in English].

11. Vapnichnyi, S., Mitsa, O., & Oryshych, S., red. (2020). *Litnia shkola z prohramuvannia [Summer school on programming]: materialy leksii, umovy ta rozbir zadach 2017-2019 rr.* Uzhhorod: Rik-U, 336 [in Ukrainian].

12. Skiena, S.S. (2020). *The Algorithm Design Manual*. Springer [in English].

13. Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2022). *Introduction to algorithms*. MIT press [in English].

14. *Zymnyaya shkola po programyrovannyu [Winter school of programming]*. Harkov: XNURE, 14-24 fevralya 2014, 294 [in Ukrainian].

15. Vapnichnyi, S.D., Putkanadze, Kh., Mitsa, O.V., & Horoshko, Yu.V. (2020). Orhanizatsiia ta analiz eksperymentu shchodo dystantsiinoho navchannia osnov prohramuvannia uchniv viddalenykh raioniv [Organization and analysis of an experiment on distance learning of basics of programming students in remote areas]. *Aktualni pytannia suchasnoi pedahohiky: tvorchist, maisternist, profesionalizm – Cbrrent issues of modern pedagogy: artist, master, professionalism: materialy Mizhnar. nauk.-prakt. konf. Kremenchuk*, 283-289 [in Ukrainian].

16. *Drugokursnyk specialnosti "Kompyuterni nauky" stav internom u londonskomu ofisi "Facebook" [A sophomore majoring in Computer Science became an intern at Facebook's London office]*. Retrieved from: <https://mediacenter.uzhnu.edu.ua/news/druhokursnyk-spetsialnosti-komp-iuterni-nauky-stav-internom-u-londonskomu-ofisi-facebook/2021-12-27-50086> [in Ukrainian].

17. *Yak uzhgorodski ajtivci Yevropu pidkoryuyut – As Uzhhorod's IT specialists conquer Europe*. Retrieved from: <https://mediacenter.uzhnu.edu.ua/news/iak-nashi-ajtivtsi-ievropu-pidkoriaiut/2020-09-03-43062> [in Ukrainian].

Received: February 21, 2022

Accepted: March 31, 2022