

Олімпіадні задачі з *інформатики*

Рубан М.М., Устінов С.Є.

Розв'язання задач
II етапу Всеукраїнської олімпіади
з інформатики — 2010 р.

Серія заснована у 2008 році

УДК 37.016:004.021

ББК 22.1

P820

Рецензенти

кандидат фіз.-мат. наук, ПАВЛЕНКО В.І.,
завідувач кафедри комп'ютерних систем та мереж
Горлівського регіонального інституту ВНЗ
ВМУРоЛ «Україна»

кандидат фіз.-мат. наук, КАДУБОВСЬКИЙ О.А.,
Слов'янський державний педагогічний університет,
доцент кафедри ГМВМ

Затверджено

Вченою радою Слов'янського державного
педагогічного університету.

Відповідальний за випуск

кандидат фіз.-мат. наук, завідувач кафедри
алгебри Рябухо О.М.

Рубан М.М., Устінов С.Є.

ОЛІМПІАДНІ ЗАДАЧІ З ІНФОРМАТИКИ: Розв'язання задач II етапу
Всеукраїнської олімпіади з інформатики – 2010 р. – Слов'янськ, 2011. –
122 с.: іл. (Серія: Викладачі СДПУ – учням, студентам, вчителям...)
ISBN 978-966-1554-60-2

Для вчителів, методистів, керівників гуртків з програмування, викладачів та студентів спеціальностей «математика», «інформатика» вищих навчальних закладів.

УДК 37.016:004.021

ББК 22.1

ISBN 978-966-1554-60-2

© Рубан М.М., Устінов С.Є.

Зміст

Вступ	4
Список позначень	6
Класифікація задач	7
Автономна Республіка Крим	9
Волинська область	13
Дніпропетровська область	19
Донецька область	27
Запорізька область	39
Луганська область	55
Львівська область	61
Полтавська область	67
Рівненська область	76
Сумська область	85
Тернопільська область	90
Харківська область	95
Хмельницька область	103
Чернівецька область	108
Задачі для самостійного розв'язку	115
Література	118
Ресурси мережі Інтернет	120

Вступ

*Есть книги, которые надо только отведасть,
есть такие, которые лучше всего проглотить,
и лишь немногие стоить разжевать и переварить...*
/Фрэнсис Бэкон (Francis Bacon)/

Одним з основних напрямків розвитку змагань з інформатики серед учнів та студентів є поширення різноманітних турнірів зі спортивного програмування. Це і міжнародні турніри, що проходять під егідою міжнародної організації АСМ (*Association for Computing Machinery*), і турніри в рамках держав та їх територіальних одиниць, змагання між студентами університетів, галузевих університетів тощо. Всі ці турніри поєднує єдина форма проведення, та спільна форма організації завдань. Такі турніри вже перешагнули границі очних змагань і все більше існує інтернет-ресурсів, за допомогою яких організуються як індивідуальні змагання так і командні (див. *Ресурси мережі Internet*).

Чудовими матеріалами для підготовки до нових олімпіад є збірники олімпіадних задач, які вже проходили. І хоча з інформатики проведено вже велику кількість олімпіад та турнірів, а з задачами, які там було запропоновано можна ознайомитись на сайтах Інтернету, систематизованих і методично оформлених збірок в навчальних закладах практично не існує.

Даний випуск містить задачі олімпіад II етапу Всеукраїнської олімпіади з інформатики, які проходили в Волинській, Дніпропетровській, Донецькій, Запорізькій, Луганській, Львівській, Полтавській, Рівненській, Сумській, Тернопільській, Харківській, Хмельницькій, Чернівецькій областях та Автономній Республіці Крим в 2010 році. І хоча для 8-9 класів умовою задачі не передбачалось написання комп'ютерної програми, але наведені приклади розв'язання задач будуть дуже корисними.

Тексти задач приведені в тому вигляді, в якому вони пропонувалися на олімпіаді, щоб зберегти повне уявлення про кожну область. До кожної задачі наведено вказівки та програмне рішення.

Основна мета посібника – допомога:

- ✓ вчителю, викладачу, тренеру – системно організувати підготовку учнів та студентів, до олімпіад і турнірів з інформатики;
- ✓ учням, студентам – перевірити рівень своєї підготовки, розв'язуючи завдання із збірника і оцінюючи свої розв'язки;
- ✓ всім учасникам олімпійського руху з інформатики – підвищити свій олімпійський потенціал, ознайомившись із запропонованими розв'язками задач.

Ми використовуємо мову *Pascal* для представлення алгоритмів, які

описуємо, тому, що це одна з найпоширеніших мов програмування. Спочатку алгоритми представляються у вигляді аналізу чи в іншій абстрактній формі. Це зроблено для того, щоб показати весь спектр проблем при розв'язку практичних задач: від проблеми формалізації задачі до проблем, які можуть виникнути під час виконання завершеної програми.

Алгоритми, які ми пропонуємо, можна реалізувати на будь-якій відомій Вам мові програмування.

Авторські задачі, які приведені в розділі «*Задачі для самостійного розв'язку*», Ви можете знайти та спробувати розв'язати в *online*-режимі на сайті *E-Olimp* (<http://www.e-olimp.com/>).

Бажаємо успіхів!

Список позначень

*Плохая система обозначений может сделать
хорошее изложение плохим, а плохое – еще худшим;
лучшее обозначение – отсутствие обозначений.
/Пол Ричард Халмош (Paul Richard Halmos)/*

<i>Позначення</i>	<i>Опис</i>
$ x $	модуль числа x
$n!$	факторіал числа ($n! = 1 \cdot 2 \cdot \dots \cdot n$, $0! = 1$)
$\sum_{i=1}^n i$	сума чисел від 1 до n
$\sum_{k=1}^n a_k$	сума елементів від a_1 до a_n
$\min(a, b)$	мінімальне число серед чисел a та b
$\max(a, b)$	максимальне число серед чисел a та b
$x = \overline{a, b}$	x пробігає всі цілі значення на відрізку $[a, b]$
$a : b$	a кратне b
$a \not\div b$	a не кратне b
$\gcd(a, b)$	найбільший спільний дільник (НСД) двох цілих чисел a та b
$\text{lcm}(a, b)$	найменше спільне кратне (НСК) двох цілих чисел a та b
$\lfloor x \rfloor$	найбільше ціле число, яке менше або рівне x : $\max\{n \mid n \leq x\}$
$\lceil x \rceil$	найменше ціле число, яке більше або рівне x : $\min\{n \mid n \geq x\}$
$\text{mod}(a, b)$	остача від ділення: $a - b[a/b]$
$\text{div}(a, b)$	цілочисельне ділення: $\lfloor a/b \rfloor$

Класифікація задач

*Есть лишь два вида языков программирования:
те, на которые вечно жалуется, и те,
которые никогда не используются.
/Бьёрн Страуструп (Bjarne Stroustrup)/*

Для зручності в користуванні задачі збірника було розподілено на класи. В дужках записано номер сторінки на якій Ви зможете знайти умову задачі.

Прості задачі

Number (9), Четвертування шоколаду (27), Вирізання числа (28), Магазин (32), Рибальська сітка (39), Ремонт (40), Бінарні числа (41), В гостях у Кролика (51), Зернини (55), Олімпіада (62), Ключ (69), Щаслива кількість (76), Різниця (103), Битва екстрасенсів (108).

Теорія чисел

Number (9), Годинник (13), FiWi (19), Вирізання числа (28), Відсутнє число (28), Суперзмагання (29), Шестикутники (30), «Комедія чи бойовик?» (63), Одиниці (67), Ділення на 3 (79), Маршрутне таксі (80), Мінімальна сума (81), Фактор числа (88), Ланцюжок непростих чисел (97), Число (100), Математика у шостому класі (105), Дільники (106), Чарівні зернята (109).

Системи числення

Binary (9), Кодування (23), Бінарні числа (41), Одиниці (67), Хитрий бобер (70), «Симпатичні числа» (90), Римська система (95).

Геометрія

Catcher (20), Четвертування шоколаду (27), Фермерське щастя (44), Стільниковий зв'язок у великому місті (46), Броунівський рух (73), Їжачок (78), Подорож (85).

Перебір варіантів

Двоякі числа (14), Дні ненародження (34), Морський бій (36), Фермерське щастя (44), Найбільший добуток (56), Номер Будинку (91), Числа (103), Дільники (106), Бізнес (113).

Комбінаторика

Шестикутники (30), Щасливі числа (49), День народження Петрика (65).

Структури даних

Каси (21), Дні ненародження (34), Найбільше середнє (37), Експеримент (43), Стільниковий зв'язок у великому місті (46), Ферзь, тура і кінь (53), Числа (103), Материк (110), Безпека банків (112).

Сортування та пошук

Кодування (23), Найбільше середнє (37), Безпека банків (112).

Моделювання

Зламаний годинник (33), Наперстки (42), Експеримент (43), Ферзь, тура і кінь (53), Петрик П'яточкін та слоненята (61), Друкарська справа (76), Вираз (86).

Пошук в глибину (рекурсія)

Знаки (24), Хитрий бобер (70), Нафтові плями (68), Фігури (83).

Динамічне програмування

Fібо (10), Станції (15), Морський бій (36), Купівля квитків (57), Мінімальна сума (81), Фішка (92), Бізнес (113).

Довга арифметика

Fібо (10), Фішка (92), Число (100).

Автономна Республіка Крим

Учитывая текущее плачевное состояние наших программ, можно сказать, что программирование определено все ещё черная магия и, пока, мы не можем называть его технической дисциплиной.
/Билл Клинтон (Bill Clinton)/

Задача 1. Number. Знайти суму цифр натурального числа N , де $1 \leq N \leq 2\,147\,483\,647$.

Вхідні дані. Вхідний файл містить число N .

Вихідні дані. Вихідний файл містить результат.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

<i>Приклад введення:</i>	<i>Приклад виведення:</i>
7259	23

Аналіз. Задача дуже проста та, на нашу думку, не потребує аналізу. Весь розв'язок зводиться до сумування останньої цифри числа (остача від ділення на 10 — $n \bmod 10$). Після сумування «позбуваємось» від останньої цифри (беремо цілу частину від ділення на 10 — $n \operatorname{div} 10$). Ці дії повторюємо до тих пір, поки не обробимо всі цифри вхідного числа.

Програмна реалізація наведеного розв'язку:

```
1 var s, n: integer;  
2 begin  
3   readln(n); s:=0;  
4   while n>0 do  
5     begin  
6       s:=s+n mod 10;  
7       n:=n div 10;  
8     end;  
9   Writeln(s);  
10 end.
```

Задача 2. Binary. Перевести число N ($1_2 \leq N \leq 1\,111\,111\,111_2$), задане в двійковій системі числення, в десяткову систему числення.

Вхідні дані. Вхідний файл містить число N , задане в двійковій системі числення.

Вихідні дані. Вихідний файл містить результат.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

<i>Приклад введення:</i>	<i>Приклад виведення:</i>
10100	20

Аналіз. Більшості олімпійцям добре відомий алгоритм переведення числа з двійкової у десяткову систему числення. Суть алгоритму полягає

в наступному: кожен цифру даного числа множимо на двійку в степені на 1 меншому, ніж порядковий номер цифри в вхідному числі (починаючи з останньої), та сумуємо всі отримані добутки. Наприклад, для числа 1100101_2 мусимо виконати наступні дії:

$$1100101_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 = 101_{10}$$

Програмна реалізація наведеного розв'язку:

```

1 var n, t, k: integer;
2 begin
3   Readln(n); t:=0; k:=1;
4   while n>0 do
5     begin
6       t:=t+k*(n mod 10);
7       n:=n div 10;
8       k:=k*2;
9     end;
10  Writeln(t);
11 end.
```

Проте перевести число з двійкової у десяткову систему можна також нестандартним алгоритмом, який потребує меншої кількості операцій (за рахунок того, що немає потреби зберігати поточну степінь двійки). Для кожної цифри вхідного числа слід виконати такі дії: помножити отриману раніше суму на 2 та додати до неї поточну цифру. В даному алгоритмі підрахунок необхідно розпочинати з найбільшого розряду числа.

Проілюструємо алгоритм на числі 1100101_2 :

$$1100101_2 = ((((((1 \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1 = 101_{10}$$

Програмна реалізація наведеного розв'язку:

```

1 var n: string;
2     t, i: integer;
3 begin
4   Readln(n); t:=0;
5   for i:=1 to Length(n) do
6     t:=t*2+ord(n[i]='1');
7   Writeln(t);
8 end.
```

Задача 3. Fibo. Послідовність Фібоначчі виглядає наступним чином:

$$1, 1, 2, 3, 5, 8, 13, 21, \dots$$

Не важко бачити, що в цій послідовності перші два числа дорівнюють 1, а всі наступні числа, починаючи з третього, дорівнюють сумі двох попередніх.

Іншими словами, послідовність Фібоначчі задається наступною рекурентною формулою:

$$F_1 = 1; F_2 = 1; F_n = F_{n-1} + F_{n-2}$$

Напишіть програму, яка знаходить N -те число Фібоначчі.

Вхідні дані. Єдиний рядок вхідного файлу містить натуральне число N ($1 \leq N \leq 10000$).

Вихідні дані. Єдиний рядок вихідного файлу повинен містити N -те число Фібоначчі.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
1	1
7	13
14	337

Аналіз. Дана задача на перший погляд здається дуже простою, проте, слід звернути увагу на обмеження $N \leq 10000$. Вже 93-ій член послідовності Фібоначчі виходить за межі типу *int64*.

$$F_{93} = 12\ 200\ 160\ 415\ 121\ 876\ 738$$

Виходячи з цього, робимо висновок, що для розв'язку задачі необхідно використовувати «довгу арифметику».

Для розв'язку реалізуємо сумування та виведення довгих чисел. Кожне довге число зберігатимемо в масиві цілих чисел з 250 елементів $dl = \text{array}[0..250]$ of *Integer*. В кожному елементі масиву зберігатимемо 9 розрядів «довгого числа», таким чином в масиві поміститься число, що складається з 2250 цифр. Зауважимо, що F_{10000} складається з 2090 цифр.

Для пошуку N -го числа Фібоначчі необхідно поступово знаходити F_i для $i = 3, 4, 5, 6, 7, \dots, N$, щоразу виражаючи $F_i = F_{i-1} + F_{i-2}$, використовуючи операцію додавання довгих чисел.

Програмна реалізація наведеного розв'язку:

```

1 type dl=array[0..250] of Integer;
2 var a,b,c:dl;
3     i,n:integer;
4     t:int64;
5     w:string;
6 begin
7     Readln(n);
8     a[0]:=1; a[1]:=1; b[0]:=1; b[1]:=1; n:=n-2;
9     while n>0 do
10    begin

```

Задачі для самостійного розв'язку

«Сядем на минутку», – предложила Аліса.

«На минутку?» – воскликнул Кролик.

«А если бы на тебя сели?»

/Льюис Кэррол (Lewis Carroll), Приключения Аліси в Стране чудес/

Задача А. Млинці з медом.

Якось Вінні-Пух та П'ятачок разом снідали млинцями з медом та смачним молоком. На столі в них було 3 тарілки з млинцями: на першій тарілці було 3 млинця, на другій – 5, а на третій 7 млинців. Щоб додатково підняти настрій під час сніданку, П'ятачок запропонував Вінні пограти в просту гру за такими правилами:

- ✓ за один раз можна з будь-якої тарілки взяти та з'їсти довільну кількість млинців більшу за 0 (якщо вони на ній є);
- ✓ програє той хто з'їсть останній млинець.

Оскільки Вінні не дуже любить програвати, допоможіть йому визначити, хто виграє в цій грі, якщо відомо скільки млинців та з якої тарілки взяв і з'їв перший гравець. Кожен гравець використовує оптимальну стратегію.

Вхідні дані. Через проміжки дано одну латинську літеру (W – якщо гру починає Вінні-Пух або P – якщо гру починає П'ятачок) та два цілих числа T та M ($1 \leq T \leq 3$, $1 \leq M \leq 7$), де T – номер тарілки з якої було взято млинці, M – кількість млинців.

Вихідні дані. Вивести «Winnie-the-Pooh» (без лапок), якщо виграє Вінні-Пух або «Piglet», якщо виграє П'ятачок. У випадку коли за вхідними даними неможливо визначити переможця, то вивести «Error».

Обмеження за часом: 0.5 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
P 3 7	Winnie-the-Pooh

Задача В. Рисове поле.

Одного разу, подорожуючи по казковій країні, Кролик в одній з долин побачив рисове поле. Поле займало всю долину, яка представляє собою прямокутник розміром $M \times N$ клітин. Кролик був дуже вражений складним устроєм цього поля. Воно складалося з окремих ділянок, кожна ділянка розташована на своєму рівні. Оскільки розміри поля дуже великі, то Кролик просить Вас допомогти йому знайти ділянки поля з найбільшою і найменшою площами.



Вхідні дані. В першому рядку вхідного файлу два числа M та N ($1 \leq M, N \leq 100$). В наступних M рядках по N чисел, розділених пропусками, йде опис карти. Кожна клітина карти містить цілі числа p_i ($0 \leq p_i < 100$), які вказують на якому рівні перебуває ділянка. Ділянка це набір клітин поля, які розташовані на одному рівні і мають з іншими клітинами цього ж рівня спільну сторону.

Вихідні дані. Вивести через пропуск два числа: максимальну і мінімальну площі ділянок.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
7 7	24 1
7 7 7 7 7 7 7	
7 5 5 5 5 5 7	
7 4 2 4 4 5 7	
7 4 2 0 4 5 7	
7 4 2 2 2 5 7	
7 4 4 4 4 5 7	
7 7 7 7 7 7 7	

Задача С. Бджолиний код.

Вінні-Пух, як великий знавець меду, дуже часто заходив у гості до бджіл (він таки знайшов «правильних» бджіл, які роблять «правильний» мед). Звичайно, бджоли дуже гостинні, проте Вінні приходив у гості навіть тоді, коли хазяїв не було вдома, тому бджоли дуже часто недораховувались одного чи двох горщиків меду, що їм не подобалось.

Щоб позбавитися від непроханого гостя, було вирішено поставити сучасну систему захисту від сторонніх – кодовий замок. Проте бджіл було дуже багато і хтось постійно забував код, довелося біля дверей почепити табличку з підказкою та ускладнити систему захисту. На табличці було написано таке:

2, 5, 9, 16, 27, 45, 74, 121, 197, ...

На сенсорному дисплеї замка був напис: «Введіть 10-ий член послідовності». Вінні-Пух два дні витратив на підрахунки, проте йому так і не вдалося потрапити за зачинені двері – як виявилось, кожні 30 хвилин завдання змінювалось і доводилось шукати вже інший член послідовності.

Сова не змогла навчити Вінні-Пуха швидко підраховувати необхідне значення, проте дала гарну ідею – захопити з собою ноутбук з програмою, яка за секунду може обчислити такий необхідний код.

Напишіть для Вінні-Пуха програму і, можливо, при зустрічі він пригостить Вас здобутим медом.

Вхідні дані. Одне ціле число $N \leq 50$.

Вихідні дані. Одне ціле число – N -ий член послідовності.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
3	9
7	74



**2, 5, 9, 16, 27, 45,
74, 121, 197, ...**

Література

- [1] Ахо А., Хопкрофт Дж., Ульман Дж. *Структуры данных и алгоритмы.* / Пер. с англ. – Москва: Вильямс, 2007. – 400 с.
- [2] Беллман Р., Дрейфус С. *Прикладные задачи динамического программирования.* – Москва: Наука, 1965. – 458 с.
- [3] Величко В.Є., Рубан. М.М., Батунина В.П., Устінов С.Є. *Олімпіадні задачі з інформатики: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2007, 2008 рр.* – Слов'янськ, 2009. – 36 с.
- [4] Величко В.Є., Рубан. М.М., Устінов С.Є., Пірус Є.М. *Олімпіадні задачі з інформатики: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2010 р.* – Слов'янськ, 2010. – 59 с.
- [5] Грэхем Р., Кнут Д., Паташник О. *Конкретная математика. Основы информатики.* М.: Мир; БИНОМ. Лаборатория знаний, 2006. – 703 с.
- [6] Долинский М.С. *Алгоритмизация и программирование на Turbo Pascal от простых до олимпиадных задач.* СПб.: Питер, 2005. – 237 с.
- [7] Иванов Б.Н. *Дискретная математика. Алгоритмы и программы.* М.: Лаборатория Базовых Знаний, 2001. – 288 с.
- [8] Кирюхин В.М. Окулов С.М. *Методика решения задач по информатике. Международные олимпиады.* – М.: БИНОМ. Лаборатория знаний, 2007. – 600 с.
- [9] Кнут Д. *Искусство программирования, в 3-х томах.* М.: Вильямс, 2000.
- [10] Кормен Т., Лейзерсон Ч., Риверст Р., Штайн К. *Алгоритмы: построение и анализ.* М.: Вильямс, 2005. – 1296 с.
- [11] Левитин Ананий В. *Алгоритмы: введение в разработку и анализ.* М.: Вильямс, 2006. – 576 с.
- [12] Меньшиков Ф. *Олимпиадные задачи по программированию.* – СПб.: Питер, 2005. – 320 с.
- [13] Окулов С.М. *Программирование в алгоритмах.* – М.: БИНОМ. Лаборатория знаний, 2004. – 341 с.
- [14] Порублев И.Н., Ставровский А.Б. *Алгоритмы и программы. Решение олимпиадных задач.* – Москва: Вильямс, 2007. – 480 с.

- [15] **Радион В.С.** *Олимпиады по информатике: задачи, решения, тесты.* – Минск: Аверсэв, 2007. – 367 с.
- [16] **Седжвик Р.** *Фундаментальне алгоритмы на С.* Киев: Диасофт, 2001. – 688 с.
- [17] **Скиена С.С, Ревилла М.А.** *Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям/ Пер. с англ.* – М: КУДИЦ-ОБРАЗ, 2005. – 416 с.
- [18] **Уорен Генри С.** *Алгоритмические трюки для программистов, исправленное издание / Пер. с англ.* – Москва: Вильямс, 2004. – 288 с.
- [19] **Шень А.** *Программирование: теоремы и задачи.* – 2-е изд., испр. и доп. – М.: МЦНМО, 2004. – 296 с.

Ресурси мережі Internet

http://acmicpc.org	Офіційний сайт ACM-ICPC.
http://e-olimp.com	Система підготовки та проведення олімпіад зі спортивного програмування.
http://acm.lviv.ua	ACM-Contester Український портал ACM-спільноти.
http://acm.mipt.ru	Олімпіади з програмування на Фізтехі.
http://acm.timus.ru	Timus Online Judge — найкрупніший в Росії архів задач з різноманітних змагань зі спортивного програмування.
http://acmsolver.org	Портал для тих кого цікавлять змагання з програмування ACM-ICPC.
http://gbprog.narod.ru	Розбір олімпіадних задач з інформатики від Михайла Густокашина.
http://ips.ifmo.ru	Російська Інтернет-школа інформатики та програмування.
http://acmp.ru	Школа програміста. Красноярський край.
http://informatik.kz	Інформатика викладання та навчання.
http://www.olimp.sc170.kharkov.ua	Зимова школа з програмування, Харківський Національний Університет Радіоелектроніки.

Микола Миколайович Рубан
асистент кафедри алгебри
Слов'янський державний педагогічний університет

Сергій Євгенович Устінов
інженер-програміст лабораторії ОТІ
Слов'янський державний педагогічний університет

ОЛІМПІАДНІ ЗАДАЧІ З ІНФОРМАТИКИ
Розв'язання задач II етапу
Всеукраїнської олімпіади з інформатики – 2010 р.

Відповідальний за випуск:

О.М. Рябухо,
кандидат фізико-математичних наук,
доцент, завідувач кафедри алгебри
Слов'янський державний педагогічний університет

Підписано до друку 11.03.2011 р.
Формат 60x84 1/16. Ум. др. арк. 7,75.
Тираж 50 прим. Зам. № 327.

Підприємець Маторін Б.І.
84116, м. Слов'янськ, вул. Г.Батюка, 19.
Тел./факс +38 06262 3-20-99. E-mail: *matorinb@ukr.net*

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців,
виготівників і розповсюджувачів видавничої продукції ДК №3141, видане Державним
комітетом телебачення та радіомовлення України від 24.03.2008 р.
