

Р.С. Шпакович

Бобер - 2008

Відповіді та вказівки до розв'язування завдань



ЗМІСТ

Р.С. ШПАКОВИЧ

БОБЕР - 2008

I. 5-7 КЛАСИ

Рівень 1

1. Перша ЕОМ
2. Кілобайт
3. Тема листа
4. Леді Лавлейс
5. Пароль
6. WWW
7. Інструкція
8. Розширення
9. Радіо
10. USB

Рівень 2

1. Презентація
2. Сума
3. Поворот
4. Правопис
5. Речення
6. Лінійка
7. Заміна
8. Перегляд
9. Три способи
10. Кошенята

Рівень 3

1. Чотири хатинки
2. Відра
3. Чотири монети
4. Ханойські вежі
5. Стрибунець
6. Взасмне віднімання
7. Робот
8. Вулиці міста
9. Гірлянда
10. Контрабанда

II. 8-9 КЛАСИ

I РІВЕНЬ

1. Перша ЕОМ
2. Батько
3. MSAFEE
4. ОС
5. Утиліти
6. Файлові системи
7. Біти
8. Інструкція
9. Маска
10. Українські ЕОМ

II РІВЕНЬ

1. [ПРЕЗЕНТАЦІЯ](#)
2. [СУМА](#)
3. [ПОВОРОТ](#)
4. [ПРАВОПИС](#)
5. [КОЛЬОРИ](#)
6. [ЛІНІЙКА](#)
7. [ЗАМІНА](#)
8. [ПЕРЕГЛЯД](#)
9. [ТРИ СПОСОБИ](#)
10. [КОШЕНЯТА](#)

III РІВЕНЬ

1. [ЧЕТВЕРТА ХАТИНКА](#)
2. [ВІДРА](#)
3. [ЧОТИРИ МОНЕТИ](#)
4. [ВІСІМ МОНЕТ](#)
5. [ХАНОЙСЬКІ ВЕЖІ](#)
6. [СТРИБУНЕЦЬ](#)
7. [ВЗАЄМНЕ ВІДНІМАННЯ](#)
8. [РОБОТ](#)
9. [ГІРЛЯНДА](#)
10. [ВУЛИЦІ МІСТА](#)

III. 10-11 КЛАСИ

I РІВЕНЬ

1. [ПЕРША ЕОМ](#)
2. [ОСТРІВ КАВИ](#)
3. [ЕЛІЗА](#)
4. [БІТИ](#)
5. [ФАЙЛИ І САЙТИ](#)
6. [ІНСТРУКЦІЯ](#)
7. [УТИЛІТИ](#)
8. [АДРЕСНА МОВА](#)
9. [САЙТ](#)
10. [ОСБОРН](#)

II РІВЕНЬ

1. [ПРЕЗЕНТАЦІЯ](#)
2. [ПОВОРОТ](#)
3. [СУМА](#)
4. [ПРАВОПИС](#)
5. [КОЛЬОРИ](#)
6. [ЛІНІЙКА](#)
7. [ЗАМІНА](#)
8. [ПЕРЕГЛЯД](#)
9. [ТРИ СПОСОБИ](#)
10. [КОШЕНЯТА](#)

III РІВЕНЬ

1. [ЧЕТВЕРТА ХАТИНКА](#)
2. [ВІДРА](#)
3. [ХАНОЙСЬКІ ВЕЖІ](#)
4. [СТРИБУНЕЦЬ](#)
5. [ВЗАЄМНЕ ВІДНІМАННЯ](#)
6. [ФАЛАНГИ](#)
7. [РОБОТ](#)
8. [ХАТИНКИ І ВЕЖІ](#)
9. [ВУЛИЦІ МІСТА](#)
10. [ГІРЛЯНДА](#)

ПЕРЕДМОВА

У цьому збірнику ви знайдете відповіді та вказівки до розв'язування всіх завдань першого конкурсу «Бобер» в Україні. Завдання для кожної вікової групи були розбиті на три рівні по 10 завдань:

1. Історія розвитку та основи інформаційно-комунікаційних технологій.
2. Робота з офісними та користувацькими програмами.
3. Основи алгоритмізації та програмування, інтерактивна реалізація класичних алгоритмів.

У вказівках до завдань першого рівня подається не лише коротка відповідь на конкретне питання, а й описуються історичні умови, в яких зароджувались та реалізовувались основні ідеї сучасної інформатики. Інформатика – це наука, яка продовжує бурхливо розвиватись і взаємодіяти з іншими науки. І навпаки, мабуть не існує такої галузі науки, яка не використовує досягнення інформатики. Численні історичні та літературні відступи робляться тут для того, щоб показати, що досягнення в сучасній інформатиці базуються і на високому рівні гуманітарної культури вчених. В шкільних підручниках недостатньо приділено уваги досягненням української інформатики. Тому у завданнях і поясненнях зроблена спроба частково заповнити цю прогалину. У захоплюючій історії зародження інформатики та ЕОМ у різних країнах, українські вчені були серед світових лідерів. Багато фактів з історії інформатики взято з численних науково-популярних праць професора Бориса Малиновського, безпосереднього учасника цих хвилюючих подій. Оскільки на вивчення офісних і користувацьких програм в шкільній програмі відводиться біля третини виділеного на інформатику часу, стільки ж відведено завдань і в нашому конкурсі.

Особлива роль відведена завданням третього рівня. На жаль, багато учнів, які чудово володіють офісними та користувацькими програмами, втрачають інтерес до вивчення основ програмування, вважаючи, що це занадто складно і нудно. Запропоновані завдання є містком, що зв'язує цікаві логічні і практичні задачі з комп'ютерними алгоритмами. Навіть якщо ви досить легко розв'язали деякі задачі, не полінуйтесь проглянути і розв'язки, запропоновані тут. Методи розв'язування, які ілюструються на простих завданнях, дозволять вам розв'язувати і складніші задачі програмування.

Якщо завдання повторюється у кількох вікових групах, використовуються гіперпосилання на відповідні пояснення для молодших класів.

Платформу, на якій реалізовано весь конкурс (від оформлення задач до їх перевірки), розробив студент Львівського національного університету ім. Івана Франка, випускник Львівського фізико-математичного ліцею Тарас Шпот. Значну допомогу у створенні цієї платформи надала компанія “Eleks Software”.

Автори завдань:

Викладачі Львівського фізико-математичного ліцею Лілія Костів та Ростислав Шпакович, студент Львівського національного університету ім. Івана Франка Тарас Кушнір.

Частину завдань взято з банку задач Міжнародної спільноти «Бобер», книг з програмування українських авторів І. М.Порубльова та А.Б.Ставровського. На фото – учасники конференції Міжнародної спільноти «Бобер», 2009 рік.



I. 5-7 класи

Рівень 1

1. Перша ЕОМ

Вважається, що перший реально діючий програмований комп'ютер був створений у 1941 році німецьким вченим Конрадом Цузе (на фото – пам'ятник на його могилі). Елементною базою були магнітні реле, програмно реалізовувались лише послідовності команд (розгалуження і цикли не програмувалися). Обчислення виконувались в двійковій системі числення, операція множення забирала 3-5 секунд. Цузе пропонував створити протягом двох років новий обчислювальний комплекс на електронних лампах, але німецьке військове командування відмовилось фінансувати цей проект.



Перша широкомасштабна ЕОМ, основною елементною базою якої були електронні вакуумні лампи, **ENIAC** (Electronic Numerical Integrator and Computer) була створена на замовлення армії США в **1946 році в Філадельфії**. Обчислення виконувались в десятковій системі числення, вага ЕОМ – 27 тон, споживана потужність 150 кВт, швидкодія – 5000 операцій додавання або 300 операцій множення за секунду. Незважаючи на використання десяткової системи числення, збільшення швидкодії внаслідок використання електронних ламп було вражаючим.



Розробники першої ЕОМ молоді американські вчені Екерт і Мочлі проглядають роздруківку результатів роботи ENIAC (1946 рік).

2. Кілобайт

1 кілобайт = 2^{10} байтів = 1024 байти.

3. Тема листа

На будь-яку електронну скриньку щодня можуть приходити сотні листів. Якщо тема не цікава отримувачу або відсутня, то швидше всього цей лист буде знищений непрочитаним.

Тому назва теми електронного повідомлення повинна бути короткою і відображати його суть.

Найкращий варіант у даному випадку – «**Домашнє завдання**».

4. Леді Лавлейс

Дочка великого англійського поета Джорджа Байрона Ада Лавлейс (1815-1852) більше 10 років співпрацювала з Чарльзом Беббіджем, який розробив схему аналітичної машини, що повинна була автоматизовано виконувати громіздкі обчислення.

Оскільки ідеї Беббіджа не знайшли розуміння в уряді Англії, він змушений був шукати спонсорів в університетах Італії і Франції. Тому конспект його лекцій був виданий спочатку французькою мовою. Після того, як Ада переклала його англійською, конструктор Беббідж попросив її написати коментарі, як математик. Примітки виявилися втричі більшими за конспект. В розумінні ролі машини вона пішла далше від автора. Вона писала програми для машини, яка могла з'явитися лише в майбутньому. Ось фрагмент з її роботи: «Аналітичну машину не можна розглядати виключно як обчислювальний пристрій. Це механізм, що дозволяє виконувати найрізноманітнішу роботу з символами. Він дозволяє встановлювати зв'язки між машинними операціями і абстрактним мисленням на високому рівні. Це нова потужна мова, яка в майбутньому може використовуватися для аналізу».

Тому в її честь названа одна з сучасних найнадійніших мов програмування **Ада**, яка використовується на різних вбудованих бортових комплексах.

5. Пароль

Електронна поштова скринька – одне з найвразливіших місць, через яке робляться спроби проникнення до конфіденційної інформації, розміщеної на комп'ютері. Зокрема пароль до неї не повинен містити інформацію, яка власнику дуже близька. Якщо хтось захоче прочитати його пошту, він теж може знати цю інформацію.

Найкращий варіант з запропонованих – **re!va8eD**.

6. WWW



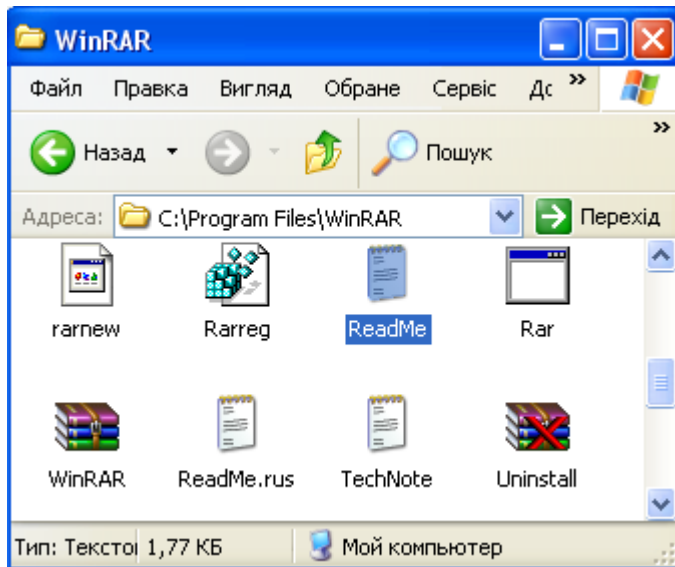
WWW – World Wide Web.

Автором цього і багатьох інших термінів сучасного Інтернету є його «батько» - британський вчений Тімоті Бернерс-Лі. У 1989 році він зумів поєднати концепцію тодішніх електронних мереж з ідеєю гіпертексту та універсальної адреси ресурсу,

давши поштовх лавиноподібному розвитку мережі Інтернет фактично у її сучасному вигляді. На фото справа – комп'ютер NeXT, який був використаний Тімом Бернерсом, як перший в світі веб-сервер. В минулому році творець WWW одержав грант у 350 тис. доларів на вирішення завдання про походження й джерела інформації в мережі. Через півроку Бернерс-Лі отримав додатковий стимул для вирішення цього завдання – він став жертвою електронних шахраїв, перевівши гроші за різдвяний подарунок у неіснуючий інтернет-магазин. Сер Бернерс-Лі заявив, що сума невелика, але знайти шахраїв – це справа честі!



7. Інструкція



У відомій книзі Льюїса Керола «Пригоди Аліси в країні чудес» головна героїня часто натикалася на різні продукти з прикріпленими записками «З'їж мене» або «Випий мене».

Тому перші пакети прикладних програм містили текстові файли з описом особливостей цих програм під назвою «Прочитай мене» («**ReadMe**»).

8. Розширення

Файл з розширенням **bat** не є графічним.. Це текстовий файл, що містить інформацію для запуску програм на машинній мові, готових до виконання (файлів з розширеннями exe і com).

9. Радіо

Правильна відповідь – **усі ці програми.**

10. USB

Навіть якщо безпечне від'єднання не виконано, сам пристрій і файли на ньому не будуть пошкоджені. Але якщо ви працювали з якимись файлами і залишили їх відкритими в оперативній пам'яті, то внесені зміни можуть не записатися на запам'ятовуючий пристрій. **При безпечному відключенні програма нагадає вам про ці файли.** Електричне живлення пристрою не відключається цією командою. Дійсно, USB-пристрої інколи виходять з ладу при від'єднанні від комп'ютера, але це, на жаль, стається і при безпечному від'єднанні.

Рівень 2

1. Презентація

Правильна відповідь – **MS PowerPoint.**

2. Сума

Нехай **a** -початкове число в комірці **B1**. Після того, як її вміст збільшили вдвічі, це значення стало рівним **2a**, сума зросла на величину **a**. Щоб сума не змінилась, **вміст комірки B2 потрібно зменшити на половину вмісту комірки B1.**

Примітка. У деяких учасників конкурсу виникало запитання:

5 – це початковий чи кінцевий вміст комірки **B1**?

Насправді це не впливає на відповідь.

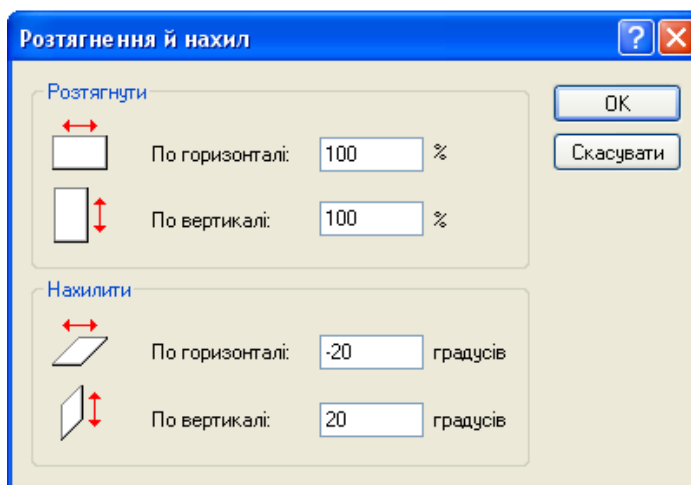
3. Поворот

Багато учасників відповіли, що такий поворот в програмі Paint неможливий. Дійсно, такої безпосередньої операції немає. Але якщо всі горизонтальні лінії малюнка нахилити по вертикалі, а вертикальні лінії – по горизонталі на однаковий кут, то отримаємо поворот всієї фігури на вказаний кут. Слід було врахувати, що в Paint додатні кути нахилу відкладаються в додатніх напрямках осей *x* та *y*. Тому для повороту ці нахили потрібно брати з протилежними знаками.

Отже, потрібно **4 операції, N = -20 градусів**:

- 1) Виділення верхнього фрагмента
- 2) Його переміщення:
- 3) Виділення всього малюнка
- 4) Його одночасний нахил на -20 градусів по горизонталі і 20 градусів по вертикалі (в програмі Paint це команда

Розтягнути/Нахилити розділу **Малюнок** головного меню):



4. Правопис

Команда перевірки правопису знаходиться в розділі головного меню **Сервіс(Tools)**.

Текстові редактори самі підкреслюють червоною хвилястою лінією слова, які можуть написані невірно. Клікаючи по такому слову правою кнопкою мишки, можна побачити інші варіанти, які пропонує редактор. Але останнє слово залишається за користувачем. У багатьох випадках його варіант правильніший.

5. Речення

Найменше місця файл з цим речення займе в **Блокноті** – 25 байтів.

В програмі Microsoft Word – 20 кілобайтів!

6. Лінійка

Команда **Лінійка** знаходиться у розділі **Вид (View)**. Використовується для швидкого задання полів, абзацних відступів та табулювання вибраного фрагменту тексту.

7. Заміна

За допомогою команди **Правка/Заміна (Edit/Replace)**.

8. Перегляд

Розділ меню **Показ слайдів (Slide Show)**.

9. Три способи

Не існує способу «З використанням конфігуратора».

10. Кошенята

Потрібно **5 операцій**. Наприклад:

- 1) Виділити всю фотографію;
- 2) Повернути її на 90 градусів:



- 3) Відобразити зверху вниз:



- 4) Скопіювати отримане зображення вправо:



5) Відобразити його зліва направо:



Рівень 3

1. Чотири хатинки

Ідея цієї задачі запозичена з однієї з кращих вітчизняних книг по олімпіадних задачах з програмування І.М.Порубльова та А.Б.Ставровського «Алгоритми та програми» (стор.83) Авторський розв'язок базується на використанні кількох допоміжних рекурсивних алгоритмів, які взаємно використовують один одного:

- перед тим, як включити потрібну лампочку виконується допоміжний алгоритм, який виключає всі лампочки зліва, крім найближчої;
- і навпаки – перед тим, як виключити лампочку, виконується допоміжний алгоритм, який залишає зліва включеною лише найближчу лампочку.

Більшість учасників успішно справились з цією задачею, використовуючи саме такі логічні міркування. Значно важче записати формальний алгоритм, що складається з кількох команд, який може просто реалізувати будь-який виконавець. Початківцям корисно спочатку навчитись записувати словесні алгоритми для себе чи для своїх товаришів, а вже потім писати програми, які зможуть виконувати комп'ютери, машини або роботи.

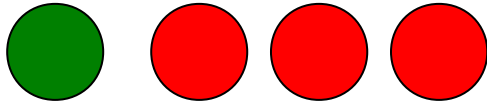
Дане завдання можна легко виконати, використовуючи такий простий циклічний алгоритм:

Поки всі лампочки не включені, повторювати дві наступні дії:

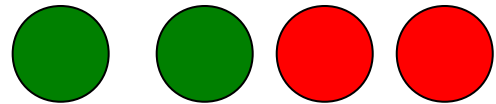
- 1) Змінити колір першої лампочки;***
- 2) Змінити колір лампочки, розташованої зразу за першою зеленою лампочкою.***

Нижче показано всі п'ять циклів, потрібних для реалізації алгоритму.

1)

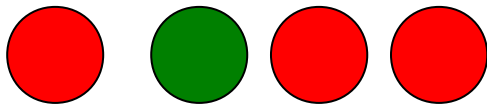


2)

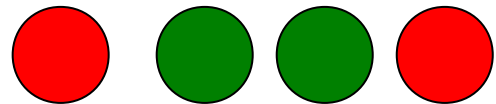


Другий цикл:

1)

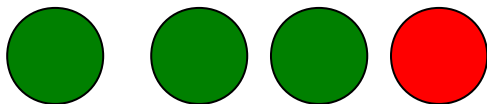


2)

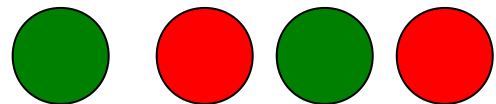


Третій цикл:

1)

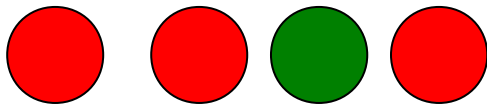


2)

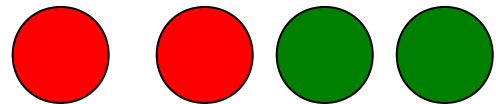


Четвертий цикл:

1)

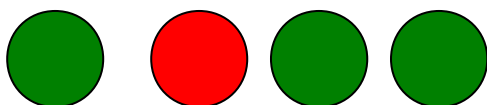


2)

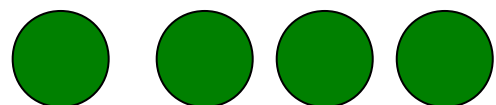


П'ятий цикл:

1)



2)



Неважко переконатись, що даний алгоритм працює для довільної кількості лампочок. Попробуйте записати аналогічний алгоритм до подібного завдання для старшокласників «Четверта хатинка».

2. Відра

Хоча задачі такого типу відомі досить давно, вони залишаються хорошим прикладом практичного застосування цілочисельного ділення. Нагадаємо, що крім звичайного ділення є ще такі операції:

div – цілочисельне ділення, **mod** – остача від цілочисельного ділення.

Наприклад: $7 \text{ div } 4 = 1$; $7 \text{ mod } 4 = 3$.

Дане завдання можна виконати двома способами:

- 1) Набирати воду семилітровим відром, а чотирьохлітрове використовувати, як мірку для отримання в семилітровому п'яти літрів. Оскільки $21 \bmod 16 = 5$, то потрібно 3 рази наповнити семилітрове відро, за шість операцій перелити з нього в чотирьохлітрове відро 16 літрів, три рази вилити воду з чотирьохлітрового відра – всього 12 операцій.
- 2) Набирати воду чотирьохлітровим відром, а семилітрове використовувати як мірку. Оскільки $12 \bmod 7 = 5$, чотирьохлітрове відро потрібно наповнити три рази, за чотири операції перелити 12 літрів в семилітрове відро і один раз вилити з нього воду – всього 8 операцій.

Попробуйте визначити, яка мінімальна кількість операцій потрібна в аналогічних задачах для старшокласників.

3. Чотири монети

Якщо під час першого зважування терези зрівноважені, це означає, що фальшива друга або четверта монета, якщо ні – то перша або третя. За друге зважування визначаємо, яка саме з монет фальшива. Відповідь: **4,2,3,1**.

4. Ханойські вежі

Історія задачі «Ханойські вежі» окутана багатьма легендами. Жак Арсак у своїй знаменитій книзі «Програмування ігор та головоломок» пише, що автором цієї задачі є викладач математики Люка ліцею Сент-Луї кінця XIX століття. Сам Люка подав цю задачу у вигляді легенди про тибетських монахів, які продовжують переставляти 50 дисків з однієї осі на іншу за вказаними правилами. Пізніше появилася легенда про монастир біля Ханоя і 64 вежі.

В інформатиці ця задача традиційно використовується як приклад рекурсивного алгоритму:

«Для того щоб переставити N кілець з початкової осі на кінцеву потрібно виконати наступні операції:

- 1) **Переставити N-1 кільце на проміжну вісь, тобто виконати алгоритм для N-1 кільця;**
- 2) **Переставити найбільше кільце на останню вісь;**
- 3) **Вважаючи проміжну вісь за початкову, а початкову за проміжну, знову виконати алгоритм для N-1 кільця».**

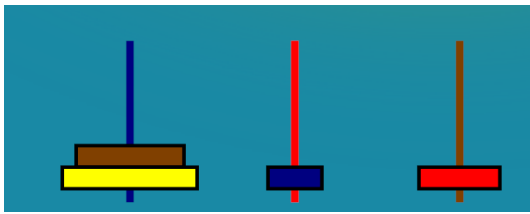
Таким чином виконання цього алгоритму полягає у багатократному виклику самого себе для кількості кілець, зменшеної на одиницю (рекурсія!). Цей алгоритм дуже зручний, щоб підрахувати загальну кількість перестановок кілець, адже саме це число потрібно знайти у задачі Люка. Але зрозуміти, яке кільце на яку вісь ставити на кожному ході, з цього алгоритму дуже важко. Виявляється, що, як і в задачі «Чотири хатинки», існує простий циклічний алгоритм розв'язання:

«1. Поки на останній осі не будуть всі кільця крім найменшого, повторювати наступні дві операції:

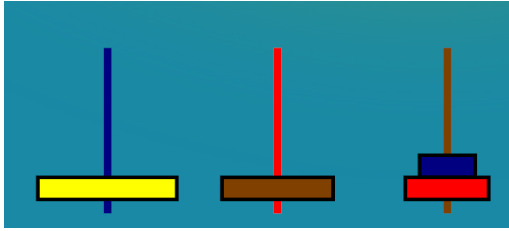
- 1) **Найменше кільце переставити на одну вісь вправо;**
{Вважати, що після останньої осі знову йде перша};
- 2) **Переставити інше кільце, крім найменшого;**
{Виявляється, що на кожному кроці можлива єдина така перестановка! Адже кільце можна ставити або на порожню вісь, або менше кільце на більше}

2. Переставити найменше кільце на останню вісь»

Положення після 1-го циклу:



Положення після 2-го циклу:



Попробуйте реалізувати цей алгоритм. Які зміни треба внести в нього, якщо кількість кілець непарна?

5. Стрибунець

Для розв'язання цього завдання найпростіше використати так званий «жадібний» алгоритм, тобто на кожному кроці робити хід, який найшвидше приводить до мети. Отже, максимально можлива довжина першого стрибка дорівнює двом і стрибунець попадає на третій біт. Максимально можлива довжина другого стрибка дорівнює трьом і стрибунець попадає на шостий біт. За третій стрибок стрибунець попадає на останній біт.

6. Взаємне віднімання

Цей спосіб знаходження НСД двох чисел, описується майже у всіх підручниках з програмування, як один з перших прикладів практичного використання циклічного алгоритму.

Розв'язок:

1155

315

1 дія: $1155 - 315 = 840$

2 дія: $840 - 315 = 525$

3 дія: $525 - 315 = 210$

4 дія: $315 - 210 = 105$

5 дія: $210 - 105 = 105$

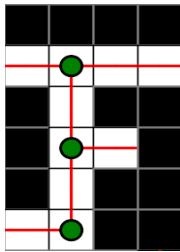
7. Робот

Відповідь – LD (наліво і вниз). Задача має ще один розв'язок – направо і вгору, але його в списку відповідей немає.

8. Вулиці міста

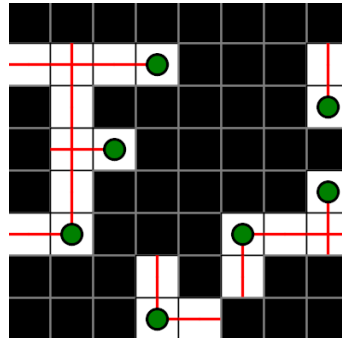
При заданих цінах камери завжди вигідніше ставити на перетинах вулиць, а не на кожному вулицю окремо. Всі вулиці можна умовно розбити на дві множини – горизонтальні та вертикальні вулиці. Тоді завдання зводиться до однієї з популярних задач сучасного програмування :

«Знайти максимальну кількість паросполучень з елементів двох множин (камер на перетинах вулиць), так щоб їх сумарна вартість була мінімальною. У кожному парі входить по одному елементу з кожної множини, кожний елемент може використовуватись лише один раз».



Розглянемо лівий верхній фрагмент нашої схеми, який складається з трьох горизонтальних і однієї вертикальної вулиці. З трьох можливих пар слід вибрати вертикальну і нижню горизонтальну вулиці, оскільки вартість камери, встановленої на їх перетині, найменша.

Мінімальна вартість покриття всіх вулиць міста камерами спостереження – 38 грошових одиниць.



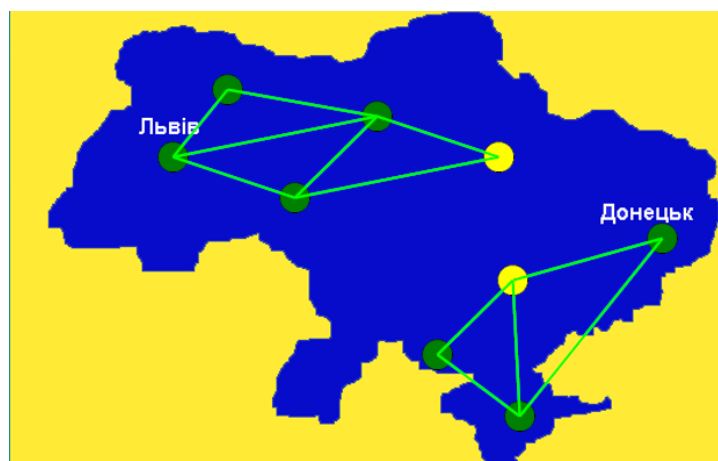
Один з можливих варіантів розміщення камер.

9. Гірлянда

Ця задача просто розв'язується **методом динамічного програмування**, суть якого полягає в тому, що **спочатку розв'язується задача для прямокутника малих розмірів. Потім розміри прямокутника поступово збільшуються, при цьому використовуються розв'язки, отримані для менших прямокутників.**

Спочатку розглядаємо нижню стрічку гірлянди. Починаючи з крайньої правої лампочки, рухаємося вліво, по черзі вмикаючи всі вимкнуті лампочки. Таким чином проходимо стрічку за стрічкою. Оскільки при ввімкненні наступної лампочки свій стан змінюють лише лампочки, розташовані вище і зліва, всі пройдені лампочки залишаються включеними.

10. Контрабанда



II. 8-9 класи

I рівень

1. Перша ЕОМ

Перша в континентальній Європі діюча ЕОМ, на якій вже в 1951 році виконувались розрахунки, необхідні для розв'язання важливих науково-технічних проблем була створена в Києві під керівництвом С.О.Лебедева. В 1950 році вона була створена, як макет великої ЕОМ і називалась **МЭСМ** (Макет Электронной Счетной Машины). Та вона зразу почала експлуатуватись і слово **Макет** у її назві поміняли на **Мала**. Швидкодія її була до 3000 операцій в хвилину, використовувалась двійкова система числення.. Програми завантажувались в оперативну пам'ять, споживана потужність 25 кВт.



Велика машина **БЭСМ-1** була створена вже в 1952 році (на фото зверху). Її швидкодія повинна була складати 10 тисяч операцій за секунду, яка була б у той час найвищою у світі. У той час йшло змагання між київськими та московськими вченими у створенні кращої ЕОМ. Міністерство машинобудування і приладобудування Радянського Союзу фактично стало на сторону москвичів і відмовилось постачати у Київ сучасні потенціалоскопи для запам'ятовуючих пристроїв. В той же час московським розробникам постачали всі необхідні пристрої.. Тому Лебедев був змушений використовувати запам'ятовуючі пристрої на ртутних трубках і швидкодія київської БЭСМ стала такою ж, як і в московської «Стрелы» – 2000 операцій за секунду.

2. Батько



Перша пограмістка Ада Лавлейс була дочкою одного з найвидатніших поетів всіх часів та народів Джорджа Байрона. Для нас він цікавий ще й тим, що першим з європейських поетів у поемі «Мазепа» звернувся до теми України та однієї з найдраматичніших подій в її історії.

3. McAfee



McAfee – одна з найбільших у світі компаній, заснована в 1989 році, яка займається розробкою антивірусних програм, програм захисту від шпигунських програм та інших загроз в Інтернеті. Її останні антивірусні програми – “McAfee Virus Scan Plus” та “3-User McAfee Internet Security”

4. ОС

Операційними системами є Linux та OS/2.

- Операційну систему OS/2 почали створювати разом компанії IBM та Microsoft. На початку 90-х років їхні шляхи розійшлися. Microsoft перетворила свою версію в Windows NT. IBM продовжувала розробляти OS/2, у якої значно менші вимоги до апаратних засобів, ніж у Windows NT.
- Операційна система Linux є найбільш поширеною вільною ОС з відкритим кодом. Її ядро у 1991 році створив фінський програміст Лінус Торвальдс на базі системи Unix. Звідси і така назва. Саме завдяки відкритості коду у вдосконаленні цієї операційної системи беруть участь сотні тисяч програмістів з усього світу. Зараз лише біля 2% коду системного ядра написані самим Торвальдсом.



Пінгвін Тукс (Торвальдс Unix) – офіційний талісман Linux.

5. Утиліти

Відповідь: А-1, Б-2, В-3, Г-4.

- 1) Іноді, після встановлення або видалення тієї чи іншої програми, система починає працювати нестабільно. Тому перед встановленням нової невідомої програми краще створити контрольну точку відновлення системи. Для цього клацніть **Пуск > Всі програми > Стандартні > Службові > Відновлення системи**. Після цього вам буде запропоновано повернути систему до попереднього стану або створити нову точку для майбутнього відновлення.
- 2) Для усунення проблем, викликаних пошкодженням секторів та іншими помилками на жорстких дисках можна використовувати програму перевірки дисків Chkdsk. Для цього потрібно:
 - У вікні **Мій комп'ютер** клікнути правою кнопкою по назві відповідного диска.
 - Вибрати команду **Властивості/ Сервіс**.
 - На панелі **Перевірка диска** натиснути кнопку **Виконати перевірку**.
- 3) Програма очищення диска надає змогу зменшити кількість непотрібних файлів на жорсткому диску. Вона видаляє тимчасові, системні та інші файли, які стали непотрібними. Для запуску програми вибрати **Пуск > Всі програми > Стандартні > Службові > Очищення диска**.
- 4) При записуванні інформації на новий жорсткий диск, кожний файл записується у послідовно розміщені кластери. Після багатократного видалення непотрібної інформації вивільнені кластери розміщені на диску хаотично. Тому нові великі файли записуються на несусідніх кластерах. Це збільшує час їх завантаження в оперативну пам'ять і призводить до зменшення часу експлуатації диска. Програма дефрагментації диска переписує всі ізольовані фрагменти файлів у сусідні кластери.

У сучасних операційних системах програма дефрагментації запускається автоматично через певні проміжки часу.

6. Файлові системи

Файлова система — це спосіб збереження інформації у вигляді файлів, який використовується операційною системою.

Файлова система FAT (*File Allocation Table*) або FAT16 – розроблена в 1977 році Біллом Гейтсом та Марком МакДональдом. Використовувалась в операційній системі MS-DOS . В операційних системах Microsoft, до Windows Millennium Edition включно, використовувалась файлова система FAT32. Числа 16 і 32 – це кількість бітів, які відводяться для зберігання номера кожного кластера. Зараз використовується файлова система NTFS (*New Technology File System*). Вона дозволяє розмежовувати доступ до інформації різних користувачів на одному комп'ютері. DOS не є файловою системою.

7. Біти

$$2^5=32<45<34=2^6$$

Отже потрібно 6 бітів.

Для швидкого переведення числа з десяткової системи в двійкову можна використати стандартну програму **Калькулятор** в інженерному виді, натиснувши функціональну клавішу **F8**.

8. Інструкція

9. Маска

Шаблону **b??r*.txt** відповідає лише файл **bebras.txt**.

В більшості пошукових програм зірочкою позначається довільна група символів, знаком запитання – рівно один довільний символ.

10. Українські ЕОМ

Основоположником українського комп'ютеробудування є Сергій Олексійович Лебедєв. Під його керівництвом була створена перша в континентальній Європі ЕОМ. Безпосередню участь в створенні цієї машини брали співробітники його лабораторії Л.Н.Дашевський та С.П.Погребинський. Академік В.М. Глушков – основоположник інформаційних технологій в Україні та колишньому СРСР. В 1964 році під його керівництвом розроблено проект Єдиної державної мережі обчислювальних центрів, тобто на два роки раніше від американського проекту АРПАНЕТ. Але радянське державне керівництво, на відміну від американського, не зрозуміло важливості цього проекту і фактично згорнуло його реалізацію. На жаль політичні лідери тоталітарних і консервативних держав часто стають перепоною на шляху науково-технічного прогресу власних народів. Згадайте нерозуміння ідей [Цузе](#) і [Бєббіджа](#) у їх країнах. В Радянському Союзі це нерозуміння супроводжувалось ще й політичними репресіями проти багатьох талановитих вчених.

II рівень

11. [Презентація](#)

12. [Сума](#)

13. [Поворот](#)

14. [Правопис](#)

15. **Кольори**

Зображення складається з 2000 пікселів. Колір кожного пікселя кодується 4 бітами. Отже, всього може використано $2^4 = 16$ різних кольорів.

16. [Лінійка](#)

17. [Заміна](#)

18. [Перегляд](#)

19. [Три способи](#)

20. [Кошенята](#)

III рівень

1. **Четверта хатинка**

Дивіться розв'язок задачі [«Чотири хатинки»](#) для 5-7 класів.

2. [Відра](#)

3. [Чотири монети](#)

4. **Вісім монет**

Розв'язок попередньої задачі використовується тут як допоміжний алгоритм.

Використання допоміжних алгоритмів є одним з основних елементів програмування.

В одній програмі часто буває необхідно передбачити можливість виконання певного допоміжного алгоритму при різних початкових умовах. Тому допоміжний алгоритм записується окремо. Величини, яким при виклику допоміжного алгоритму з основної програми, можуть надаватись різні конкретні значення, називаються формальними параметрами. Самі конкретні значення називаються фактичними параметрами.

В умові задачі на нижній блок-схемі зображено допоміжний алгоритм знаходження за два зважування фальшивої монети серед монет з формальними номерами I,J,K,L. В головному алгоритмі у першому зважуванні визначається, у якій половині знаходиться фальшива монета – з першої по четверту монети, чи з п'ятої по восьму. Після цього формальним параметрам I,J,K,L надаються фактичні значення 1,2,3,4 або 5,6,7,8 відповідно і виконується допоміжний алгоритм.

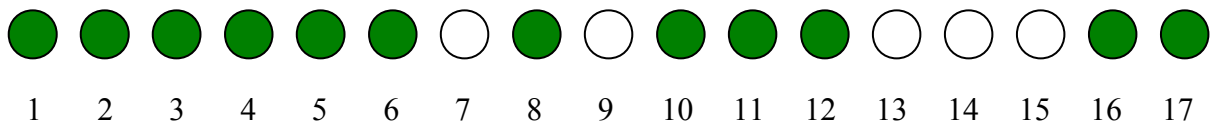
Перше зважування: $W[1]+W[2]=W[3]+W[4]$

5. [Ханойські вежі](#)

6. [Стрибунець](#)

Відповідь – 4 стрибки.

Але, на жаль, жадібний алгоритм не завжди дозволяє отримати розв'язок. Розглянемо таку бітову доріжку:



Згідно запропонованого жадібного алгоритму маршрут мав би бути таким:

1=>3=>6=>10... Наступний стрибок неможливий, оскільки 13-й, 14-й і 15-й біти заборонені. Знайдіть правильний маршрут.

Повний алгоритм розв'язування цієї задачі можна подивитись у книзі Іллі Порубльова «Олімпіадні завдання з інформатики», звідки вона і запозичена.

7. [Взаємне віднімання](#)

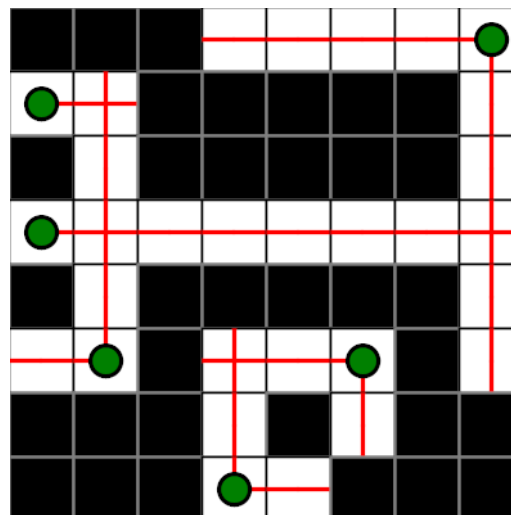
8. [Робот](#)

9. [Гірлянда](#)

10. [Вулиці міста](#)

Як і [в аналогічній задачі для 5-7 класів](#), потрібно вибрати максимальну кількість паросполучень горизонтальних та вертикальних вулиць таким чином, щоб сумарна вартість розташованих на них камер спостереження була мінімальною. Оскільки є 4 вертикальні вулиці та 6 горизонтальних, то може бути лише 4 паросполучення різних вулиць. На двох горизонтальних вулицях, які залишились, потрібно поставити окремі найдешевші камери.

Мінімальна вартість покриття всіх вулиць міста камерами спостереження – 34 грошові одиниці.



Один з можливих варіантів розміщення камер.

III. 10- 11 класи

I рівень

1. Перша ЕОМ

В 1946 році один з творців [першої ЕОМ](#) Джон Мочлі прочитав курс лекцій по принципах побудови ЕОМ в Пенсильванському університеті. Серед слухачів був молодий вчений Моріс Уїлкс з Кембриджського університету, саме того, в якому 100 років тому Чарльз Беббідж запропонував проект програмованої обчислювальної машини. Повернувшись до Англії, талановитий вчений зумів за дуже короткий строк (в 1949 році) створити машину **EDSAC** (Electronic Delay Storage Automatic Computer). Це була **перша в Європі працююча ЕОМ** і перша в світі ЕОМ, в якій програма зберігалась в оперативній пам'яті. Машина працювала в двійковій системі. В США таку машину створили лише через два роки.



На фотографії 1998 року зліва направо: творець першої в Європі ЕОМ, почесний професор Академії Наук України **Моріс Уїлкс**, головний конструктор першої вітчизняної цифрової напівпровідникової керуючої машини широкого призначення, автор багатьох цікавих книг по історії інформатики, професор **Борис Малиновський**, розробник першого світі бізнес-комп'ютера, професор **Френк Ленд** (Великобританія).

2. Острів кави

Ява – це мова програмування, відомий кавовий бренд та острів в Індонезії – найбільш густонаселений регіон в світі. Площа в 5 разів менша від площі України, проживає близько 90 млн. осіб. Об'єктно орієнтована мова програмування Java, випущена компанією Sun Microsystems d 1995 році. Одна з головних переваг мови – незалежність програм від апаратної частини та операційної системи, встановленої на комп'ютері.



3. Еліза

Eliza – це знаменита комп'ютерна програма, написана в 1966 році Джозефом Вейзенбаумом. Це перша програма, яка перетворила діалог користувача з ЕОМ в ілюзію людського спілкування. Програма названа в честь Елізи Дулітл, героїні п'єси «Пігмаліон» Бернарда Шоу, яку навчали мові людей «вищого світу». На фото – « діалог» Елізи (Н. Сумська) та Генрі Хігінса (А. Хостікоєв) у виставі театру ім. Івана Франка.



4. Біти

$2^{10}=1024$, отже з використанням 10 бітів у двійковій системі числення можна представити 1024 різних числа.

5. Файли і сайти

Переглядати web-сторінки та працювати з файлами та папками можна за допомогою програми **Internet Explorer**.

6. [Інструкція](#)

7. [Утиліти](#)

8. Адресна мова

Катерина Логвинівна Ющенко – один з авторів Адресної мови. Свій науковий шлях вона розпочинала у Львові (1946-1950 роки). Потім переїхала в Київ, написала перші програми для першої континентальної ЕОМ, створеної під керівництвом С.О.Лебедєва. В зв'язку з обмеженістю ресурсів і різноманітністю архітектури перших ЕОМ, для кожної з них писались окремі програми. Катерина Ющенко створила в інституті кібернетики ім. Глушкова всесвітньо відому наукову школу теоретичного програмування. Адресна мова – перше фундаментальне досягнення цієї школи. Вона підготувала появу мов програмування з використанням непрямої адресації.



За схожість наукових біографій Катерину Ющенко називають українською леді Лавлейс.

9. Сайт

Сайт – це множина зв'язаних веб-сторінок.

10. Осборн

Osborne 1 – перший в світі портативний переносний комп'ютер (ноутбук).

Ідея створення ноутбука, як і лазерного принтера та багатовіконного інтерфейсу, належить співробітнику компанії Херох Алану Кею (1968 р.). В 1979 році Вільям Могридж створив на замовлення NASA комп'ютер Grid Compass (на фото зліва), який використовувався на космічних кораблях. Лише відсутність автономного живлення не дозволило зробити його переносним.

Через два роки Адам Осборн спроектував перший, дійсно переносний, комп'ютер Osborne 1. Він важив 11 кг, коштував 1800 доларів. Дисплей підтримував лише текстовий режим: 24 стрічки по 52 символи. Користувався величезним попитом, за місяць продавалось до десяти тисяч комп'ютерів.



II рівень

1. [Презентація](#)
2. [Поворот](#)
3. **Сума**
В дужках повинен бути вираз (**A1:A4;A6**).
4. [Правопис](#)
5. [Кольори](#)
6. [Лінійка](#)
7. [Заміна](#)
8. [Перегляд](#)
9. [Три способи](#)
10. [Кошенята](#)

III рівень

1. [Четверта хатинка](#)
2. [Відра](#)
3. [Ханойські вежі](#)
4. [Стрибунець](#)
Відповідь – 5 стрибків.
5. [Взаємне віднімання](#)
6. **Фаланги**

В задачі потрібно знайти НСД трьох чисел тим же способом, що і в попередній задачі, але за найменшу кількість дій віднімання. Виявляється, що для чисел 252, 336 та 441, як і в задачі «Стрибунець» можна використати жадібний алгоритм. Його суть полягає в тому, що у кожній дії потрібно від найбільшого числа віднімати середнє по величині число, тобто якнайшвидше прирівняти всі три числа:

1 дія: $C=441-336=105$;

2 дія: $B=336-252=84$;

3 дія: $A=252-105=147$ і т. д.

Всього потрібно **8 дій**, НСД=21.

Але, на жаль, жадібний алгоритм не завжди дозволяє отримати найшвидший розв'язок. Наприклад, застосуємо цей алгоритм для чисел 16, 18 та 20:

1 дія: $C = 20 - 18 = 2$;

2 дія: $B = 18 - 16 = 2$;

3 дія: $A = 16 - 2 = 14$ і т. д.

Всього буде виконано 9 дій. Але існує алгоритм, який дозволяє прирівняти всі три числа до НСД за сім дій віднімання. Попробуйте знайти цей алгоритм.

7. Робот

Відповідь: LUR.

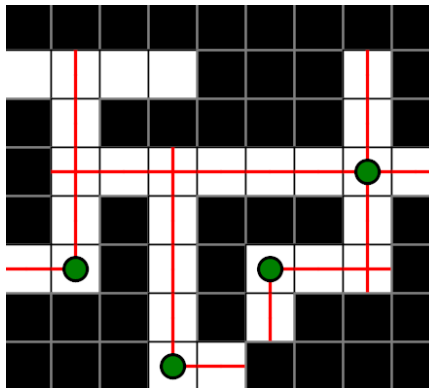
8. Хатинки і вежі

З пояснень до задач «Четверта хатинка» та «Ханойські вежі» ви вже знайомі з простими словесними алгоритмами їх розв'язання. Виявляється, що між розв'язками цих задач існує повна аналогія, згідно якої правильному кліканню у першій задачі відповідає правильний хід у другій задачі:

Якщо K непарне, то переставити кільце з номером K на одну вісь вправо, якщо парне – то на дві.

9. Вулиці міста

Як і в [аналогічній задачі для 5-7 класів](#), потрібно вибрати максимальну кількість паросполучень горизонтальних та вертикальних вулиць таким чином, щоб сумарна вартість розташованих на них камер спостереження була мінімальною. Оскільки є 4 вертикальні вулиці та 5 горизонтальних, то може бути лише 4 паросполучення різних вулиць.



Оптимальний варіант розміщення камер на перехрестях.

Крім того, на верхній горизонтальній вулиці потрібно поставити найдешевшу камеру. Мінімальна вартість покриття всіх вулиць міста камерами спостереження – 31 грошова одиниця.

10. Гірлянда