II (районний) етап Всеукраїнської учнівської олімпіади з інформатики

Київ, 13 листопада 2016 р.

Максимальна оцінка за кожну з чотирьох задач — 100 балів.
Для всіх задач обмеження на час — 1 секунда / тест; обмеження на пам'ять — 256 МБ.

Матеріали олімпіади буде оприлюднено на сайті **kievoi.ippo.kubg.edu.ua**, а також на **soi.org.ua**.
Автор задач — Данило Мисак. Художник — Олександра Десятерик.

---

1. Карти (назва програми: `cards.cpp` / `cards.pas` / `cards.*`)

Якось, граючи у карти, Нетямко помітив, що в нього на руці вісім різних карт червової масті: є всі карти від двійки до десятки за винятком однієї. Допоможіть Нетямку визначити, якої саме карти бракує.

Вхідні дані
У вхідному файлі у довільному порядку задано вісім різних натуральних чисел у межах від 2 до 10.

Вихідні дані
У вихідний файл виведіть натуральне число у межах від 2 до 10 — карту червової масті, якої бракує.

Приклад

Вхідний файл `cards.in`	Вихідний файл `cards.out`
6 3 10 9 2 8 4 5	7

Історична довідка
Гральні карти мають тисячолітню історію. Їх було винайдено в Китаї, а перша згадка датується IX століттям та описує «гру у листя».

2. Хрестики-нулики (назва програми: `ttt.cpp` / `ttt.pas` / `ttt.*`)

У хрестики-нулики грають на полі 3×3 двоє гравців, що ходять по черзі. Перший гравець ставить хрестик у довільну клітинку поля; його суперник ставить нулик у будь-яку іншу клітинку; перший гравець ставить ще один хрестик у будь-яку незайняту клітинку; далі його суперник ставить нулик і т. д. Виграє той, після чийого ходу деякі три хрестики або три нулики стоятимуть на одній горизонталі, на одній вертикалі чи на одній з двох діагоналей поля. Якщо всі клітинки поля вже зайнято, але жодні три хрестики або нулики не займають однієї горизонталі, вертикалі чи діагоналі, вважають, що гра завершилась унічию.

Якось Нетямко грав у хрестики-нулики з товаришем і замислився: чи не час закінчувати поточну партію, а якщо так, то хто виграв? Допоможіть хлопцю знайти відповіді на ці питання.

Вхідні дані
Вхідний файл складається з трьох рядків, у кожному з яких міститься по три символи з набору: **`x`** (мала латинська літера *x*, що позначає хрестик), **`o`** (мала латинська літера *o*, що позначає нулик), **`.`** (крапка, що позначає порожню клітинку). Символи пробілами не розділено. Вхідні дані задають коректну позицію гри у хрестики-нулики (можливо, й початкову, коли ще не зроблено жодного ходу).

Вихідні дані

У вихідний файл виведіть єдиний символ (не забувши про перенесення рядка):

- **x** — малу латинську літеру *x*, — якщо гра вже закінчилася і виграли хрестики;
- **o** — малу латинську літеру *o*, — якщо гра вже закінчилася і виграли нулики;
- **=** — знак рівності, — якщо гра закінчилася внічию;
- **.** — крапку, — якщо гра ще не закінчилася.

Приклади

Вхідний файл **ttt.in**	Вихідний файл **ttt.out**
..x ox. x.o	x
ooo .x. xx.	o
x.x xoo oxo	.

Історична довідка

Існує припущення, що хрестики-нулики походять зі Стародавнього Єгипту. В усякому разі у дуже схожу гру під назвою Terni Lapilli грали у I ст. до н. е. у Римській імперії, на підтвердження чому по всьому Риму залишилися сліди накреслених крейдою ігрових полів.

3. Бики та корови (назва програми: **bac.cpp** / **bac.pas** / **bac.***)

Правила гри в бики й корови такі. Гравець задумує довільне чотирицифрове (від 1000 до 9999) число, усі цифри якого різні; суперник намагається задумане число відгадати, висуваючи гіпотези: він поступово називає різні чотирицифрові числа, що також не містять однакових цифр. На кожну гіпотезу суперника гравець, що задумав число, повинен відповісти — вказати кількість угаданих суперником цифр: ті вгадані цифри, що стоять на правильних місцях, називаються *биками*, а ті, які є в задуманому числі, але стоять на інших позиціях, називаються *коровами*. Наприклад, якщо задумано число 7183, а названо 8123, то гравець, що задумував число, відповість «два бики та одна корова» (два бики — цифри 1 і 3, що стоять на своїх місцях, а корова — цифра 8, що стоїть не там, де треба).

Якось, граючи з товаришем, Нетямко спитав у нього про n чисел, ретельно записав відповіді про кожне з них і був уже близький до перемоги, але зненацька його записи розсипалися й Нетямко заплутався, яка відповідь товариша відповідала якому названому Нетямком числу. Спираючись на переплутані записи, допоможіть Нетямку вгадати, яке число задумав його товариш.

Вхідні дані

У першому рядку вхідного файлу вказано натуральне число n. У наступних n рядках міститься по одному чотирицифровому числу, про яке Нетямко питав у товариша (всі числа різні). У наступних n рядках записано по два цілих числа — відповіді товариша: перше число — кількість биків, а друге число — кількість корів у

деякому з чисел, про які питав Нетямко. Зверніть увагу, що набір з n чисел відповідає набору з n відповідей у деякому *переплутаному* порядку.

Вихідні дані

У вихідний файл виведіть задумане товаришем Нетямка число. Відомо, що це число можна відновити однозначно і воно не збігається з жодним із чисел, заданих у вхідному файлі.

Приклад

Вхідний файл **bac.in**	Вихідний файл **bac.out**
3 4712 8796 3261 2 0 0 3 1 2	1726

Коментар до прикладу

Назвавши число 4712, Нетямко дістав відповідь «один бик і дві корови».
Назвавши число 8796, Нетямко отримав відповідь «два бики і жодної корови».
Назвавши число 3261, Нетямко дістав відповідь «жодного бика і три корови».
Потім рядки в його записах переплуталися.

Історична довідка

Походження гри в бики та корови не встановлене, але відома ця гра вже щонайменше протягом ста років.

4. Вісім (назва програми: **eight.cpp** / **eight.pas** / **eight.***)

Гра у вісім проходить на дошці 3×3, де розміщено у довільному порядку 8 плиток, позначених числами від 1 до 8 (кожне число трапляється рівно по разу), а дев'ята клітинка — вільна.

6	7	2
1		3
5	4	8

Мета гри — пересуваючи плитки, досягти того, щоб вони розмістилися в порядку зростання номерів зліва направо зверху вниз, а вільна клітинка розташувалася при цьому у правому нижньому куті:

1	2	3
4	5	6
7	8	

На кожному кроці пересувати у вільну клітинку можна лише сусідні з нею плитки: перекладати плитки заборонено!

Нетямко випадковим чином виклав на дошці початкову конфігурацію плиток і хотів уже було почати гру, але замислився: а раптом головоломка вимагає великої кількості пересувань плиток, а то й зовсім не підлягає вирішенню? Бо якщо так, то він навряд чи зможе її розв'язати... Допоможіть хлопцю зекономити час та

розумові зусилля: за заданим початковим розташуванням плиток визначте, за яку найменшу кількість кроків плитки можна викласти у правильному порядку, або встановіть, що досягти цієї мети неможливо.

Вхідні дані

У вхідному файлі міститься опис початкової позиції гри: три рядки файлу містять по три цілих числа від 0 до 8, де числа від 1 до 8 позначають плитки з відповідними номерами, а число 0 — порожню клітинку. Числа не повторюються.

Вихідні дані

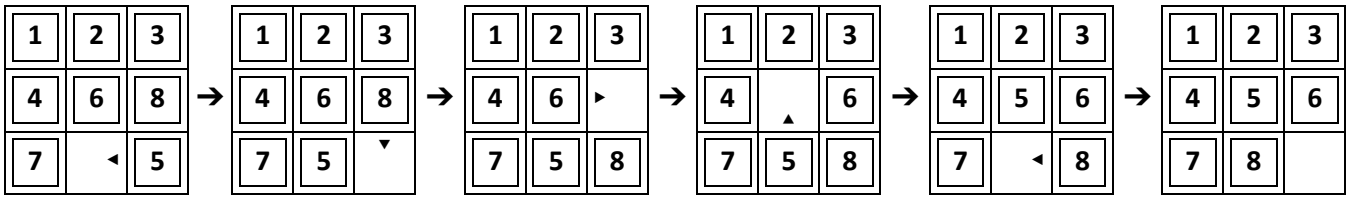
У вихідний файл виведіть найменшу кількість кроків, за які з початкової позиції можна одержати потрібну. Одним кроком ми вважаємо пересування однієї плитки на одну позицію: зсувати в одному напрямку водночас дві сусідніх плитки не можна! Якщо початкова позиція збігається з кінцевою, виведіть `0`, а якщо бажану позицію одержати шляхом пересувань плиток неможливо, виведіть `-1`.

Приклад

Вхідний файл **`eight.in`**	Вихідний файл **`eight.out`**
1 2 3 4 6 8 7 0 5	5

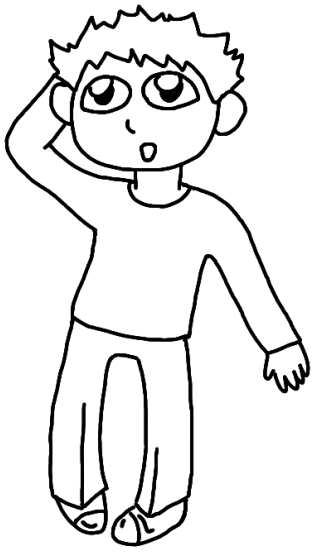
Коментар до прикладу

Досягти потрібного розташування плиток з початкового за 5 кроків можна у спосіб, показаний нижче:



Історична довідка

Гру у вісім (а точніше її варіант на дошці 4×4) було винайдено у 1870-х роках у США. Американський шахіст Сем Лойд, авторству якого інколи помилково приписують гру, запропонував велику грошову винагороду за розв'язання однієї з початкових позицій головоломки. Та ба — позиція виявилася нерозв'язною.



Ідеї розв’язання

1. Карти

Задачу можна розв’язувати у різноманітні способи. Один з найпростіших такий: відняти від числа $54 = 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$ суму всіх чисел, записаних у вхідному файлі. Різниця і буде пропущеним числом.

2. Хрестики-нулики

Необхідно перевірити всі вісім ліній поля на предмет того, чи містять вони три хрестики або три нулики. Якщо це так, вивести відповідний символ і завершити виконання алгоритму. Якщо ні, то з’ясувати, чи залишилася на полі принаймні одна вільна клітинка (гра ще триває) або такої клітинки нема (гра завершилася внічию).

Розгляд восьми ліній можна реалізувати безпосереднім «ручним» перебором або ж певним чином оптимізувати. Наприклад, написати окремі функції перевірки горизонталей, вертикалей та діагоналей, у які передавати номер відповідної лінії. Інший варіант — кожну лінію розглядати як пару з її середньої клітинки та «зсуву» відносно неї бокових клітинок. Маємо чотири лінії, що проходять через центральну клітинку поля, і чотири лінії, що проходять по його краях. Середні клітинки та зсуви обох цих наборів ліній можна задати за допомогою формул, що дозволяє зручно їх запрограмувати (див. авторський розв’язок).

3. Бики та корови

Щоб розв’язати задачу, достатньо перебрати всі чотирицифрові числа, які міг задумати товариш Нетямка, і вибрати з них те, для якого набір відповідей на висловлені гіпотези буде з точністю до порядку збігатися із заданим у вхідному файлі.

Щоб з’ясувати, чи два набори чисел збігаються з точністю до порядку, можна обидва ці набори відсортувати й далі порівняти поелементно. Так само порівнюються і набори пар чисел: їх можна впорядкувати, наприклад, за кількістю биків, а в разі рівності цього показника — за кількістю корів. Утім, ще ефективнішим буде підхід, що використовується у сортуванні підрахунком: оскільки пари кількостей биків та корів можуть набувати лише одного з 13 видів ($0/0$, $0/1$, $0/2$, $0/3$, $0/4$, $1/0$, $1/1$, $1/2$, $1/3$, $2/0$, $2/1$, $2/2$, $3/0$), ми просто порахуємо кількість пар кожного з цих видів і порівняємо між собою отримані кількості для двох наборів. А щоб рахувати було зручніше, відповідь «b биків і c корів» ототожнимо з числом $5b + c$, яке назвемо *індексом* такої відповіді. Тоді можливим парам відповідатиме проміжок індексів від 0 до 15 (пропуск індексів 9, 13 і 14 не є при цьому істотним).

4. Вісім

Задача розв’язується пошуком у ширину: вершини графа — позиції гри, а ребро між двома вершинами-позиціями наявне тоді й лише тоді, коли з однієї з цих позицій можна перейти в іншу за один крок.

Облік позицій можна вести по-різному. Один з підходів — ототожнити позицію з дев’ятицифровим (або восьмицифровим) числом, яке вийде, якщо прочитати числа на плитках зліва направо зверху вниз — наприклад, позиція, яку необхідно отримати, матиме тоді вигляд $123\,456\,780$. Але в такому випадку індексацію доведеться вести або за допомогою якої-небудь нетривіальної структури даних (як-от дерево), або скориставшись контейнером map бібліотеки STL мови C++. Інший варіант — кожну позицію розглядати як перестановку дев’яти чисел (цифр від 0 до 8) і ототожнювати з порядковим номером відповідної перестановки (числом від 0 до $9! - 1 = 362\,879$), якщо розглядати перестановки в порядку від тих, що утворюють менші числа, до тих, що утворюють більші числа. Щоб відновити за певною перестановкою її порядковий номер, знайдемо кількість перестановок, «менших» за дану: це сума кількості всіх перестановок, що починаються на цифру, меншу за першу цифру даної; тих, що починаються на ту саму цифру, але мають меншу другу цифру; тих, що мають однакові перші дві цифри, але меншу третю, і т. д.