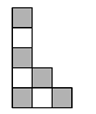
**А. Дивні шахи**

|  |  |
| --- | --- |
| **Input file name:** | chess.in |
| **Output file name:** | chess.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан нещодавно придумав свою версію шахів, в якій гра відбувається на дошці, що має форму відмінну від традиційної.

Його дошка складається з *N* стовпців, *i*-ий з яких містить *Ai* клітинок. Нижні клітинки всіх стовпців утворюють один горизонтальний ряд, причому довжини стовпців впорядковані зліва направо по незростанню. На малюнку нижче наведений приклад дошки, в якій три стовпчика містять 5, 2 і 1 клітинку відповідно.



Сьогодні Степана зацікавило питання: як розставити мінімальну кількість тур на його дошці так, щоб кожну клітинку поля била хоча б одна тура. Тура б’є ті клітинки, які розташовані з нею на одній вертикалі або одній горизонталі.

**Формат вхідних даних.** Перший рядок вхідного файлу містить ціле число *N* (1 ≤ *N* ≤ 1000) – кількість стовпців дошки. Наступний рядок містить *N* цілих чисел *A*1, *A*2, …, *AN* – кількість клітинок у стовпцях (1 ≤ *Ai* ≤ 1000, *A*1 ≥ *A*2 ≥ … ≥ *AN*).

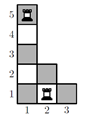
**Формат вихідних даних.** У першому рядку виведіть число *K* – мінімальну кількість тур, яку можна розставити на дошці так, щоб кожну клітинку дошки била хоча б одна тура. Наступні *K* рядків повинні містити опис позицій тур, по одній у кожному рядку. Позиція тури задається двома числами: номер стовпця, в якому стоїть тура, і номер клітинки в стовпці. Стовпці нумеруються починаючи з 1 зліва направо, клітинки в стовпцях нумеруються знизу вгору також починаючи з 1.

Якщо існує кілька розстановок, що задовольняють умову, дозволяється вивести будь-яку з них.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **chess.in** | **chess.out** |
| 3  5 2 1 | 2  1 5  2 1 |

Малюнок до прикладу:



**B. Анаграми**

|  |  |
| --- | --- |
| **Input file name:** | anagrams.in |
| **Output file name:** | anagrams.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Два непорожні рядки однакової довжини називаються анаграмами один одного, якщо другий рядок складений із символів першого, і кожен символ використовується тільки один раз. Так, пари рядків «дереза» і «резеда» є анаграмами, а пари рядків «каток» і «відкат», «стежка» і «пірат» - ні.

Ви повинні визначити, чи є два даних рядки анаграмами один одного. Рядки містять тільки символи латинського алфавіту, причому великі та малі літери вважаються різними.

**Формат вхідних даних.** Вхідний файл описує групу тестів, що складається з декількох рядків. Перший рядок файлу містить ціле число *K* (2 ≤ *K* ≤ 5) – кількість тестів у групі. Далі слідують *K* пар рядків – кожна пара відповідає одному тесту. Довжина одного рядка не перевищує 3000 символів (у 50% груп тестів ця довжина не перевищує 200).

**Формат вихідних даних.** Вихідний файл містить єдиний рядок з *K* чисел, відокремлених одним пропуском. Кожне число відповідає одному тесту і має дорівнювати 1, якщо введені рядки є анаграмами, і 0 в іншому випадку.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **anagrams.in** | **anagrams.out** |
| 4  Acad  cAda  AbRa  arBA  duda  adua  termo  metro | 1 0 0 1 |

**C. Гра «70368744177664»**

|  |  |
| --- | --- |
| **Input file name:** | game.in |
| **Output file name:** | game.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан дуже зрадів запровадженим вимушеним канікулам, адже тепер він має змогу витратити весь свій вільний час на підготовку до олімпіади з інформатики. Сьогодні Степан вирішив розібратися з двійковою системою числення. Як відомо, в ній необхідно вміти виконувати різного роду операції зі степенями двійки. Саме для вдосконалення таких навичок, у безмежних просторах Інтернету хлопець знайшов цікаву гру, назва якої «70368744177664».

Правили гри полягають в наступному. Велике квадратне поле розділене на квадратики розміром 1×1, у деяких з них знаходяться числа – степені двійки. Гравець може обрати два довільних однакових числа, після чого ці числа зникають, а на полі з’являється інше число, що рівне сумі обраних чисел.

Вдосталь награвшись в цю гру, Степан написав програму, що за початковим набором чисел на полі, знаходить найбільше число, що може з’явитися під час гри. А чи вдасться Вам повторити досягнення Степана?

**Формат вхідних даних.** У першому рядку записане одне число *N* (1 ≤ *N* ≤ 216) – кількість чисел. Другий рядок містить *N* цілих чисел *Ai* (1 ≤ *Ai* ≤ 230) – числа, що записані на полі на початку гри.

**Формат вихідних даних.** У єдиному рядку виведіть одне число – найбільше число, що може з’явитися на полі під час гри.

**Приклади вхідних та вихідних даних:**

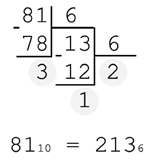
|  |  |
| --- | --- |
| **game.in** | **game.out** |
| 2  4 4 4 8 | 16 |
| 9  4 4 4 4 4 4 4 4 4 | 32 |

**D. Цікаве число**

|  |  |
| --- | --- |
| **Input file name:** | numbers.in |
| **Output file name:** | numbers.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан на факультативі з програмування почав вивчати системи числення. На першому уроці вчитель розповів про систему числення з основою два, дуже популярною в комп'ютерному світі. На другому уроці Степан дізнався про систему числення з основою три. І так далі на кожному наступному уроці він дізнавався про нові системи числення, так що на *i*-му уроці була розглянута система числення з основою *i*+1.

Щоб краще запам'ятати, Степан на кожному уроці бере одне і те ж число *X* і записує його в зошит в останній вивченій системі числення. Приклад переведення числа 81 в систему числення з основою 6:



Одного разу Степан помітив, що у записаного ним числа *X* в новій системі числення всі цифри однакові. До того ж, він розуміє, що таке відбувається вперше, і ні на якому з попередніх уроків число не виходило таким цікавим.

Повернувшись вражений додому, Степан забув про те, яку систему числення в цей день він розглядав на уроці. Допоможіть йому знайти систему числення з мінімальною основою, в якій це число має однакові цифри.

**Формат вхідних даних.** Єдиний рядок вхідного файлу містить одне ціле число *X* (1 ≤ *X* ≤ 1012) – число записане в десятковій системі числення.

**Формат вихідних даних.** Вихідний файл повинен містити одне ціле число *B* (2 ≤ *B*) – шукана система числення.

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **numbers.in** | **numbers.out** |
| 3 | 2 |
| 219 | 8 |
| 1009 | 1008 |

***Пояснення до прикладів:***

1. «3» – це «11» в системі числення з основою 2.
2. «219» – це «333» в системі числення з основою 8.
3. «1009» – це «11» в системі числення з основою 1008.

**Е. Хрестики-нулики 2015**

|  |  |
| --- | --- |
| **Input file name:** | game.in |
| **Output file name:** | game.out |
| **Time limit:** | 300 ms |
| **Memory limit:** | 128 M |

Два брата Сергій і Іван любили довгими зимовими вечорами грати в різні ігри. Особливою популярністю у хлопців користувалася гра «Хрестики-нулики». Однак через свою простоту вона їм швидко стала нецікавою. Брати вирішили придумати модифікацію цієї гри і назвали її «Хрестики-нулики 2014».  
Для гри необхідний аркуш паперу з ***N*** клітинками по вертикалі і ***N*** по горизонталі, який назвемо ігровим полем. Для ускладнення і неповторності гри деяку кількість клітин може бути закрашено. Гравці ходять по черзі, починаючи з гравця, що грає хрестиками. За один хід гравцеві дозволяється вибрати будь-яку порожню незафарбовану клітинку і намалювати там фігуру свого типу (гравець, який грає хрестиками, може малювати тільки хрестики, який грає нуликами - тільки нулики). Гра продовжується до тих пір, поки на полі існує хоча б одна порожня і незафарбована клітинка або з'явиться рядок/стовпчик, в якому відсутні порожні і незафарбовані клітинки і кількість фігур одного типу більше кількості фігур іншого типу. Якщо в цьому рядку/стовпці кількість хрестиків більше ніж кількість нуликів, то виграв гравець, який грає хрестиком, а якщо більше нуликів - виграв гравець, який грає нуликами. Якщо на полі не залишилося жодної порожньої клітинки і в кожному рядку і стовпці кількість хрестиків і нуликів збігається, то вважається, що гра зіграна в нічию. У будь-яку незафарбовані клітинку дозволяється ставити не більше однієї фігури.  
Івану дуже не щастило - він весь час програвав своєму супернику, але, тим не менш, не сумував, граючи партію за партією. Одного разу, зігравши в нічию, Іван був на сьомому небі від щастя і вирішив поділитися своїм успіхом з батьками. Для більшої переконливості хлопці вирішили показати їм ігрове поле, отримане після гри.  
Але от невдача... Іван ненавмисно пролив на ігрове поле склянку води, від якої деякі фігури розпливлися, і їх неможливо було розпізнати. Чітко можна було розібрати тільки зафарбовані клітини. Хлопці вирішили відновити ігрове поле. Допоможіть Сергію та Івану відновити ігрове поле, отримане після гри в «Хрестики-нулики 2014», яка завершилася в нічию.

**Формат вхідних даних**: Перший рядок вхідного файлу містить два цілих числа ***N, M (2 ≤ N, M ≤ 80)***.

У наступних ***N*** рядках файлу записано по ***M*** символів без пробілів. Кожен ***j***-й символ в ***i***-му рядку описує відповідну клітинку ігрового поля.

Для опису клітинки ігрового поля допускається чотири типи символів: 'X' (ASCII 88) - зафарбована клітинка.  
'+' (ASCII 43 ) - у клітинці знаходиться хрестик.

'-' (ASCII 45 ) - у клітинці знаходиться нулик.

'?' (ASCII 63 ) - у клітинці знаходиться фігура, тип якої неможливо розпізнати (тобто в клітинці може знаходитися як хрестик, так і нулик).

**Формат вихідних даних:** Вихідний файл повинен складатися з ***N*** рядків по ***М*** символів у кожному - відновлене ігрове поле. Гарантується, що рішення завжди існує. Якщо існує декілька варіантів рішення, то виведіть будь-яке з них.

***Оцінювання: 1 ≤ N, M ≤ 5*** - 15 балів Ігрове поле містить тільки символи '?' i 'X' - 40 балів

**Приклади вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **Input in game.in** | **Output in game.out** |
| 2 2  ??  ?? | +-  -+ |
| 3 4  +?XX  ?X?X  X??X | +-XX  -X+X  X+-X |
| 6 4  X??X  X?-X  ?+??  -???  ?XX+  ?XX? | X-+X  X+-X  ++--  --++  -XX+  +XX- |