

## 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  $A_i$  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 кожна клітинку поля біла 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  $N$  ( $1 \leq N \leq 1000$ ) – кількість стовпців дошки. Наступний рядок містить  $N$  цілих чисел  $A_1, A_2, \dots, A_N$  – кількість клітинок у стовпцях ( $1 \leq A_i \leq 1000, A_1 \geq A_2 \geq \dots \geq A_N$ ).

**Формат вихідних даних.** У першому рядку виведіть число  $K$  – мінімальну кількість тур, яку можна розставити на дошці так, щоб кожна клітинку дошки біла 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	2
5 2 1	1 5
	2 1

Малюнок до прикладу:



## В. 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  $K$  ( $2 \leq K \leq 5$ ) – кількість тестів у групі. Далі слідує  $K$  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	1 0 0 1
Acad	
cAda	
AbRa	
arBA	
duda	
adua	
termo	
metro	

## С. Гра «70368744177664»

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

Степан дуже зрадів запровадженням вимушених 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  $1 \times 1$ , 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  $N$  ( $1 \leq N \leq 2^{16}$ ) – кількість чисел. Другий рядок містить  $N$  цілих чисел  $A_i$  ( $1 \leq A_i \leq 2^{30}$ ) – 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	16
4 4 4 8	
9	32
4 4 4 4 4 4 4 4 4	

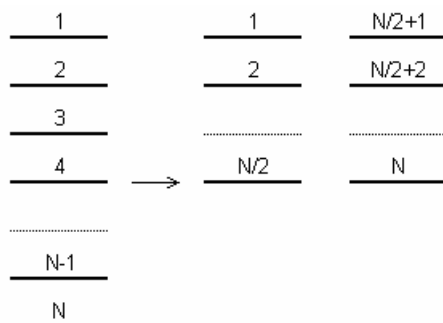
## D. Степан тестувальник

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

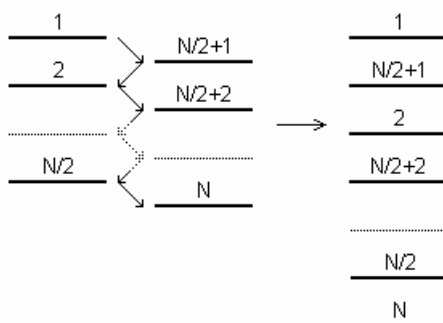
Степан влаштувався на нову роботу тестувальником – і цікаво, і зарплатня гарна. Він займається тестуванням спеціальних механізмів, які використовуються для перетасовки стопок карток.

Механізми, які тестує Степан, здійснюють перетасовку стопок з  $N$  карток, де  $N$  – парне натуральне число. Алгоритм, за якими вони працюють, полягає у застосуванні до заданої стопки певної послідовності перетворень, кожне з яких має один з 2-х типів –  $U$  або  $D$ .

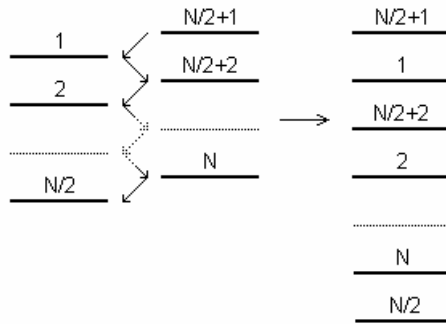
$U$ -перетворення проводиться наступним чином. Спочатку задана стопка з  $N$  карток ділиться на дві частини, перша з яких складається з  $N/2$  верхніх карток, а друга – з  $N/2$  нижніх:



Потім у результуючу стопку по черзі поміщається по одній картці з двох частин, починаючи з першої:



$D$ -перетворення відрізняється від  $U$ -перетворення тільки тим, що на другому кроці результуюча стопка починає формуватися, починаючи не з першої частини, а з другої:



Степан проводить тестування наступним чином. Він бере  $N$  карток, пронумерованих числами від 1 до  $N$ , і формує з них стопку так, щоб номери карток в стопці зростали при їх перегляді зверху вниз. Потім він поміщає стопку в машину і виконує її перетасування. Після цього Степан дістає з результуючої стопки  $K$ -у зверху картку і залежно від її номера робить висновок про справність або несправність тестованого механізму.

Для прискорення процесу тестування Степану потрібна програма, яка обчислює, чому дорівнює номер  $K$ -ї зверху картки у результуючій стопці, якщо механізм працює коректно.

**Формат вхідних даних.** Перший рядок вхідного файлу містить цілі числа  $N$  і  $K$  ( $1 \leq K \leq N \leq 2000000000$ , число  $N$  – парне), розділені пропуском. У другому рядку записано від 1 до 1000 символи «U»/«D» без пропусків. Ці символи описують послідовність перетворень, застосованих механізмом для перетасовки карток. Символ «U» відповідає  $U$ -перетворенню, а символ «D» -  $D$ -перетворенню.

**Формат вихідних даних.** Єдиний рядок вихідного файлу має містити одне ціле число – номер  $K$ -ї зверху картки у результуючій стопці.

**Приклади вхідних та вихідних даних:**

testing.in	testing.out
20 7 DUUUDUDUDU	1

## Е. Цікаве число

<b>Input file name:</b>	numbers.in
<b>Output file name:</b>	numbers.out
<b>Time limit:</b>	100 ms
<b>Memory limit:</b>	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:

$$\begin{array}{r} 81 \overline{) 6} \\ \underline{78} \phantom{0} \\ 3 \phantom{0} \overline{) 13} \\ \underline{12} \phantom{0} \\ 1 \phantom{0} \overline{) 2} \\ \underline{1} \phantom{0} \\ 1 \phantom{0} \end{array}$$

$$81_{10} = 213_6$$

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  $X$  ( $1 \leq X \leq 10^{12}$ ) – число записане в десятковій системі числення.

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

**Приклади вхідних та вихідних даних:**

<code>numbers.in</code>	<code>numbers.out</code>
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.