

Задача А. Тура

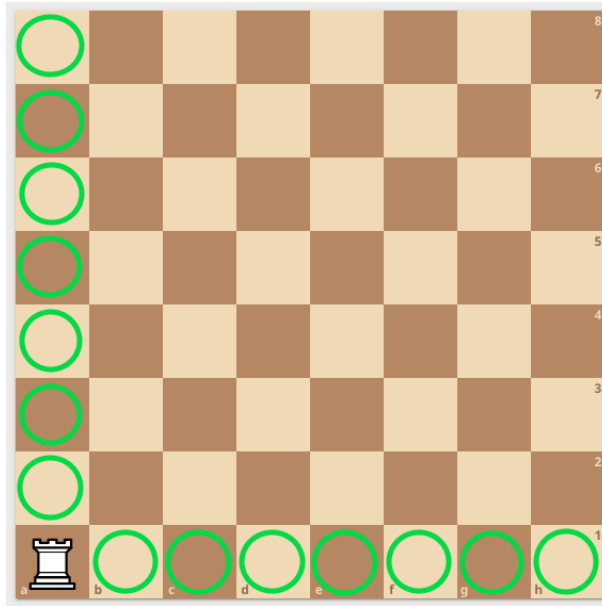
Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Дано шахівниця розміром $n \times m$. Тобто з n рядками та m стовпчиками.

У цій шахівниці є лише одна фігура — тура. Вона знаходиться у нижньому лівому куті. Більше ніяких фігур немає.

Нагадаємо, що тура за один хід може переміститися на будь-яку додатню кількість клітин по горизонталі або вертикалі, але не по діагоналі.

Знайдіть кількість клітин, на які тура може переміститися за рівно один хід.



На малюнку зображена традиційна шахівниця розміру 8×8 . У ній тура може переміститися на всі клітини, які помічені зеленим. Таких всього 14, тому відповідь 14.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 20$).

Другий рядок містить одне ціле число m ($1 \leq m \leq 20$).

Формат вихідних даних

Виведіть кількість клітин, на які тура може переміститися за один хід.

Приклади

standard input	standard output
8 8	14
3 2	3

Примітка

Пояснення, чому до першого прикладу відповідь 14, можна побачити на малюнку вище.

У другому прикладі відповідь 3, бо тура може переміститися лише на одну позицію вправо та на дві позиції вгору.

Задача В. Координати

Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Дано точку (x, y, z) у 3D-просторі.

Знайдіть **квадрат** відстані від цієї точки до центру координат (тобто, до точки $(0, 0, 0)$).

Нагадаємо, що відстань між двома точками (x_1, y_1, z_1) та (x_2, y_2, z_2) визначається за формулою

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Формат вхідних даних

Перший рядок містить одне ціле число x ($-100 \leq x \leq 100$).

Другий рядок містить одне ціле число y ($-100 \leq y \leq 100$).

Третій рядок містить одне ціле число z ($-100 \leq z \leq 100$).

Формат вихідних даних

Виведіть одне ціле число.

Приклад

standard input	standard output
1 -3 5	35

Примітка

В першому тесті нас цікавить квадрат відстані від точки $(1, -3, 5)$ до $(0, 0, 0)$.

Підставляємо в формулу координати та отримуємо:

$$\begin{aligned} & \left(\sqrt{(1-0)^2 + (-3-0)^2 + (5-0)^2} \right)^2 = \\ & = \left(\sqrt{1 + 9 + 25} \right)^2 = 35 \end{aligned}$$

Задача С. Приготуйте домашнє завдання

Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Сакурако готується до свого наступного уроку програмування, який починається сьогодні о 14 : 00, але їй потрібно завершити незакінчене домашнє завдання. Урок починається через a хвилин, а повне виконання домашнього завдання займе b хвилин. На щастя, Сакурако працювала над домашнім завданням c хвилин вчора.

Ваше завдання — визначити, чи може Сакурако завершити залишок домашнього завдання до початку уроку.

Наприклад, припустимо, що урок починається через 5 хвилин, домашнє завдання Сакурако потребує 6 хвилин для виконання, а вона вже працювала над ним 2 хвилини. У такому випадку їй потрібно $6 - 2 = 4$ хвилини, щоб закінчити його. Отже, якщо вона почне готуватися до уроку одразу, вона закінчить своє домашнє завдання (і навіть залишиться з однією додатковою хвилиною).

Формат вхідних даних

Єдиний рядок містить три цілі числа a , b та c ($0 \leq a, b, c \leq 360$).

Формат вихідних даних

Виведіть "YES", якщо у Сакурако є шанс закінчити домашнє завдання до початку уроку, або "NO" в іншому випадку.

Система оцінювання

Ви отримаєте певну кількість балів, якщо ваше рішення працює правильно для $c = 0$; тобто, Сакурако не працювала над домашнім завданням вчора.

Приклади

standard input	standard output
5 6 2	YES
5 8 3	YES
5 8 1	NO
0 0 0	YES

Примітка

Перший приклад був пояснений у легенді.

У другому прикладі урок починається через 5 хвилин. Сакурако потрібно $8 - 3 = 5$ хвилин, щоб завершити виконання домашнього завдання. Отже, вона ідеально вписується у свій графік.

У четвертому прикладі урок вже почався, але домашнє завдання вимагає нуль хвилин для виконання (чи це було усне домашнє завдання?).

Задача D. М'ячі та контейнери

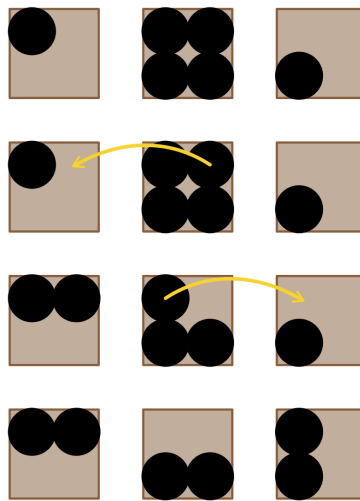
Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Сакурако — молода програмістка, яка працює над проектом у сфері робототехніки.

Вона побудувала три розумні контейнери, кожен з яких спочатку має a , b та c м'ячів відповідно. Контейнери призначені для передачі м'ячів один одному, і її завдання — написати програму, яка зробить кількість м'ячів у всіх трьох контейнерах однаковою.

Сакурако може наказати непорожньому контейнеру витягти один м'яч і передати його іншому. Однак їй цікаво дізнатися, чи можливо взагалі збалансувати контейнери, і якщо так, то скільки рухів це займе. Чи можете ви допомогти їй розробити ефективне рішення?

Припустимо, у неї є три контейнери; перший контейнер має 1 м'яч, другий контейнер має 4 м'ячі, а третій контейнер має 1 м'яч. Нижче ви можете знайти ілюстрацію для такого прикладу.



Жовті стрілки показують м'яч, який ми беремо, і куди ми його кладемо. У цьому випадку ми виконуємо дві операції.

Формат вхідних даних

Єдиний рядок містить три цілі числа a , b та c ($0 \leq a, b, c \leq 10^8$).

Формат вихідних даних

Якщо неможливо зробити кількість м'ячів у всіх трьох контейнерах однаковою, виведіть "-1".

В іншому випадку виведіть мінімальну кількість рухів, необхідних для балансування м'ячів між контейнерами.

Система оцінювання

Ви отримаєте щонайменше 25 балів, якщо ваше рішення працює правильно для $b = 0$ та $c = 0$; тобто, другий і третій контейнери порожні.

Ви отримаєте щонайменше 25 балів, якщо ваше рішення працює правильно для $b = 2 \cdot a$ та $c = 3 \cdot a$.

Приклади

standard input	standard output
1 4 1	2
2 2 3	-1
0 0 0	0
12 0 0	8

Примітка

У другому прикладі можна показати, що неможливо розподілити кулі між контейнерами, як описано в умові.

У третьому прикладі всі контейнери порожні, що означає, що у всіх них однакова кількість куль; отже, вихідні дані дорівнюють нулю.

У четвертому прикладі ми можемо дотримуватися стратегії, описаної нижче:

$(12, 0, 0) \rightarrow (11, 1, 0)$

$(11, 1, 0) \rightarrow (10, 1, 1)$

$(10, 1, 1) \rightarrow (9, 2, 1)$

$(9, 2, 1) \rightarrow (8, 2, 2)$

$(8, 2, 2) \rightarrow (7, 3, 2)$

$(7, 3, 2) \rightarrow (6, 4, 2)$

$(6, 4, 2) \rightarrow (5, 4, 3)$

$(5, 4, 3) \rightarrow (4, 4, 4)$

що займає у нас рівно 8 операцій. Ми можемо довести, що немож-

ливо збалансувати контейнери менш ніж за 8 операцій.

Задача Е. Сакурако запізналася

Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Ще один день у школі: ще один раз Сакурако запізналася!

Сьогодні вона проспала, і їй потрібно дістатися до школи якнайшвидше.

Є n пішохідних переходів, які відокремлюють Сакурако від її школи, кожен з яких має один світлофор. Кожен світлофор або зелений, або червоний. Колір кожного світлофора змінюється кожну хвилину.

Сакурако не дуже швидка, тому їй потрібно йти одну хвилину, щоб перейти один пішохідний перехід. Також вона дуже законослухняна, тому не перейде перехід, якщо світло червоне. Нарешті, вона **завжди** перейде вулицю, якщо світло зелене.

Визначте, скільки мінімального часу потрібно Сакурако, щоб дістатися до школи.

Формат вхідних даних

Перший рядок містить одне ціле число n ($1 \leq n \leq 10^6$), яке позначає кількість пішохідних переходів, які Сакурако потрібно перейти.

Другий рядок містить рядок s з n символів, які представляють початковий колір кожного світлофора; кожен символ або "G" (зелений), або "R" (червоний). Перший символ позначає колір найближчого світлофора, а останній символ позначає колір найдалшого світлофора.

Формат вихідних даних

Виведіть в одному рядку одне ціле число, яке позначає мінімальний час, необхідний Сакурако для переходу через переходи.

Система оцінювання

Ви отримаєте щонайменше 30 балів, якщо ваше рішення працює правильно для $s_1 = s_2 = \dots = s_{n-1} = s_n$; тобто всі кольори однакові.

Ви отримаєте щонайменше 30 балів, якщо ваше рішення працює правильно для $s_i \neq s_{i+1}$ ($1 \leq i \leq n - 1$); тобто всі сусідні кольори різні.

Приклади

standard input	standard output
5 RGRRG	7
10 GGRRGRGRRG	13

Примітка

У першому прикладі Сакурако може рухатися наступним чином:

Час 0:

| * ● ● ● ● |

Сакурако знаходиться на позиції 0.

Час 1:

| * ● ● ● ● |

Сакурако знаходиться на позиції 0 і починає рухатися через 1-ий пішохідний перехід.

Час 2:

| ● * ● ● ● |

Сакурако знаходиться на позиції 1 і починає рухатися через 2-ий пішохідний перехід.

Час 3:



Сакурако знаходиться на позиції 2 і починає рухатися через 3-ій пішохідний перехід.

Час 4:



Сакурако знаходиться на позиції 3.

Час 5:



Сакурако знаходиться на позиції 3 і починає рухатися через 4-ий пішохідний перехід.

Час 6:



Сакурако знаходиться на позиції 4 і починає рухатися через 5-ий пішохідний перехід.

Час 7:



Сакурако пройшла всі 5 переходів.

У другому прикладі можна показати, що мінімальний час, необхідний Сакурако для переходу через всі переходи, становить 13.

Задача F. Красивий рядок

Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Бінарний рядок — це рядок, який містить лише “0” та “1”.

Підрядок рядка — це рядок, який можна отримати, видаливши певну кількість символів з початку та/або кінця.

Сакурако вважає бінарний рядок красивим, якщо кількість одиниць щонайменше 2 і це число ділить довжину рядка.

Наприклад, “1001”, “011100” та “010100100” є красивими рядками, але рядки “101”, “01010” та “1001000” не є красивими.

Сакурако любить красиві рядки, але їй погано вдається їх шукати. Тому вона просить вас допомогти їй знайти будь-який красивий підрядок заданого рядка.

Формат вхідних даних

Перший рядок містить одне ціле число n ($3 \leq n \leq 2 \cdot 10^5$) — довжина рядка.

Другий рядок містить один бінарний рядок — рядок, в якому Сакурако втратила свій красивий рядок.

Гарантовано, що в рядку є принаймні три символи “1”.

Можна довести, що відповідь завжди існує.

Формат вихідних даних

Виведіть два цілі числа l та r ($1 \leq l \leq r \leq n$), де підрядок, що починається з l і закінчується в r , є красивим. Якщо є кілька рішень, виведіть будь-яке.

Система оцінювання

Ви отримаєте не менше 32 балів, якщо ваш розв'язок буде коректно працювати для $n \leq 1000$.

Приклади

standard input	standard output
5 10101	1 4
6 101101	3 4

Примітка

У першому прикладі підрядок “1010” є красивим, оскільки довжина, яка дорівнює 4, ділиться на кількість одиниць, яка дорівнює 2.

У другому прикладі підрядок “11” є красивим, оскільки довжина, яка дорівнює 2, ділиться на кількість одиниць, яка дорівнює 2.

Задача G. Гарний масив

Обмеження часу: 1 second
Обмеження використання пам'яті: 256 megabytes

Масив a називається красивим, якщо всі $a_i - a_{i-1}$ ($2 \leq i \leq n$) є різними.

Наприклад, $[1, 3, 2, 9]$ і $[8, 4, 6, 9]$ є красивими, а $[1, 2, 3, 9]$ і $[1, 5, 5, 9]$ — ні.

Послідовність цілих чисел b_1, b_2, \dots, b_n називається перестановкою послідовності c_1, c_2, \dots, c_n , якщо кількість кожного унікального числа у цих масивах однакова. Наприклад, $[1, 5, 3, 4, 2]$ є перестановкою $[5, 2, 1, 4, 3]$, а $[1, 3, 3, 5]$ є перестановкою $[3, 5, 1, 3]$. Проте $[1, 3, 4]$ не є перестановкою $[2, 3, 4]$ (вона не містить 2, які повинні з'явитись у послідовності, і має 1, які не повинні з'явитись).

Сакурако отримала масив. Вона хоче знайти приклад красивої перестановки заданого масиву.

Формат вхідних даних

Перший рядок містить одне ціле число n ($2 \leq n \leq 2 \cdot 10^5$) — довжина масиву.

Другий рядок містить n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — елементи масиву.

Гарантовано, що всі числа в масиві різні. Також гарантовано, що відповідь існує.

Формат вихідних даних

Виведіть гарний масив (який є перестановкою початкового масиву) в одному рядку. Якщо є кілька рішень, виведіть будь-яке.

Система оцінювання

Ви отримаєте щонайменше 16 балів, якщо ваш розв'язок спрацює правильно для $n \leq 10$.

Ви отримаєте не менше 48 балів, якщо ваш розв'язок буде правильним для $n \leq 1000$.

Приклади

standard input	standard output
5 1 2 3 4 5	3 5 4 1 2
3 4 2 3	4 2 3

Примітка

У першому прикладі масив $[3, 5, 4, 1, 2]$ є гарним, оскільки якщо ми візьмемо $a_i - a_{i-1}$ для всіх i , отримаємо:

- $5 - 3 = 2$;
- $4 - 5 = -1$;
- $1 - 4 = -3$;
- $2 - 1 = 1$.

Як ми бачимо, всі числа різні.

У другому прикладі масив $[4, 2, 3]$ є гарним, оскільки якщо ми візьмемо $a_i - a_{i-1}$ для всіх i , отримаємо:

- $2 - 4 = -2$;
- $3 - 2 = 1$.

Як ми бачимо, всі числа різні.

Задача Н. Сакурако та Різдвяний масив

Обмеження часу: 1.5 seconds
Обмеження використання пам'яті: 256 megabytes

Сакурако нещодавно прикрасила масив так, що він став Різдвяним масивом. Він був **святковим**, поки Чєфір не прийшов і не зіпсував його.

Масив довжини n називається **святковим**, якщо виконується наступне:

- для кожного i ($1 \leq i \leq n$) $a_i = a_{n-i+1}$;
- i є елемент, який зустрічається строго більше ніж $\frac{n}{2}$ разів.

Вам дано масив a довжини n , де n парне. Вам потрібно визначити мінімальну кількість елементів, які потрібно замінити, щоб цей масив став **святковим**.

«Замінити» означає вибрати будь-яке i ($1 \leq i \leq n$) та присвоїти $a_i = x$, де x — будь-яке ціле число (навіть якщо число x не зустрічалось в масиві).

Формат вхідних даних

Перший рядок містить одне ціле число n ($2 \leq n \leq 5 \cdot 10^5$; n парне), яке позначає довжину масиву. Другий рядок містить n цілих чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$).

Формат вихідних даних

Виведіть в одному рядку одне ціле число, яке позначає мінімальну кількість елементів, які потрібно замінити, щоб a став **святковим** масивом.

Система оцінювання

Ви отримаєте принаймні 10 балів, якщо ваше рішення працює правильно для $a_i = a_{i+1}$ для кожного ($1 \leq i < n$).

Ви отримаєте принаймні 20 балів, якщо ваше рішення працює правильно для $a_i = a_{n-i+1}$ для кожного i ($1 \leq i \leq n$).

Ви отримаєте принаймні 30 балів, якщо ваше рішення працює правильно для $n \leq 1000$.

Ви отримаєте додаткові 40 балів, якщо ваше рішення працює правильно без будь-яких обмежень.

Приклади

standard input	standard output
6 2 6 4 6 2 9	3
6 3 7 2 4 9 3	3

Примітка

Оптимальний набір замін в 1 прикладі:

$[2, 6, 4, 6, 2, 9] \rightarrow [2, 2, 6, 6, 2, 2]$ Для цього масиву обидві умови виконуються:

- для кожного i ($1 \leq i \leq n$) $a_i = a_{n-i+1}$
- ціле число 2 зустрічається більше ніж $\frac{6}{2} = 3$ рази в нашому масиві.

Ми можемо показати, що це мінімальна кількість операцій, які потрібно виконати.

В прикладі 2 оптимальна заміна:

$[3, 7, 2, 4, 9, 3] \rightarrow [3, 4, 4, 4, 4, 3]$

Можна показати, що це найменша кількість операцій, які потрібно виконати.