

## Задача А. Паралелепіпед

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `1 second`  
Ліміт використання пам'яті: `256 megabytes`

Вам дано розміри паралелепіпеда, знайдіть його об'єм.

### Формат вхідних даних

Перший рядок містить одне ціле число  $n$  ( $1 \leq n \leq 10$ ) — висота паралелепіпеда.  
Другий рядок містить одне ціле число  $m$  ( $1 \leq m \leq 10$ ) — ширина паралелепіпеда.  
Третій рядок містить одне ціле число  $k$  ( $1 \leq k \leq 10$ ) — довжина паралелепіпеда.

### Формат вихідних даних

Виведіть одне ціле число — об'єм паралелепіпеда.

### Приклад

<code>standard input</code>	<code>standard output</code>
2	24
3	
4	

### Зауваження

Об'єм паралелепіпеда рівний  $2 \cdot 3 \cdot 4 = 24$ .

## Задача В. Внуки

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: `1 second`  
Ліміт використання пам'яті: `256 megabytes`

У Козака Вуса рівно  $x$  синів. У кожного з його синів рівно  $(x + 1)$  син. А у кожного з його внуків рівно  $(x + 2)$  сини.

Визначте, скільки правнуків у Козака Вуса?

### Формат вхідних даних

Перший рядок містить одне ціле число  $x$  ( $1 \leq x \leq 100$ ).

### Формат вихідних даних

Виведіть одне ціле число — відповідь на задачу.

### Приклад

<code>standard input</code>	<code>standard output</code>
2	24

### Зауваження

У Козака Вуса 2 сини. У кожного сина по 3 сини, тобто у Козака Вуса 6 внуків. У кожного внука по 4 сини, тобто у Козака Вуса 24 правнуки.

## Задача С. Серце, Розчарування

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	1 second
Ліміт використання пам'яті:	256 megabytes

Сакурако має два масиви  $a_1, a_2, \dots, a_n$  та  $b_1, b_2, \dots, b_m$ . Використовуючи ці масиви, вона намалювала картину в рамці розміром  $n \times m$ . Колір клітинки  $(i, j)$  на картині визначається значенням  $c_{i,j} = a_i + b_j$ .

Її малюнок вийшов настільки захоплюючим, що його виставили в престижній художній галереї.

Одного дня ви відвідали цю галерею. Бувши допитливим програмістом, ви вирішили вивчити "красу" картини Сакурако. Ви визначаєте красу картини як площу найбільшої області в сітці, де всі клітини мають один і той же колір і є з'єднаними.

Дві клітини  $(p_1, q_1)$  і  $(p_2, q_2)$  вважаються частиною однієї з'єднаної області, якщо існує послідовність клітин  $(p_1, q_1), (x_1, y_1), \dots, (x_k, y_k), (p_2, q_2)$  така, що:

- Усі клітини в послідовності одного кольору.
- Кожна пара сусідніх клітин у послідовності є сусідніми по стороні.

Ваше завдання — визначити «красу» картини Сакурако — площу найбільшої з'єднаної області, де всі клітини мають один і той же колір.

### Формат вхідних даних

Перший рядок містить два цілі числа  $n$  та  $m$  ( $1 \leq n, m \leq 2 \cdot 10^5$ ) — довжини  $a$  та  $b$  відповідно.

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

Третій рядок містить  $m$  цілих чисел  $b_1, b_2, \dots, b_m$  ( $1 \leq b_i \leq 10^9$ ) — елементи  $b$ .

### Формат вихідних даних

Виведіть єдине ціле число — «красу» картини Сакурако.

### Система оцінювання

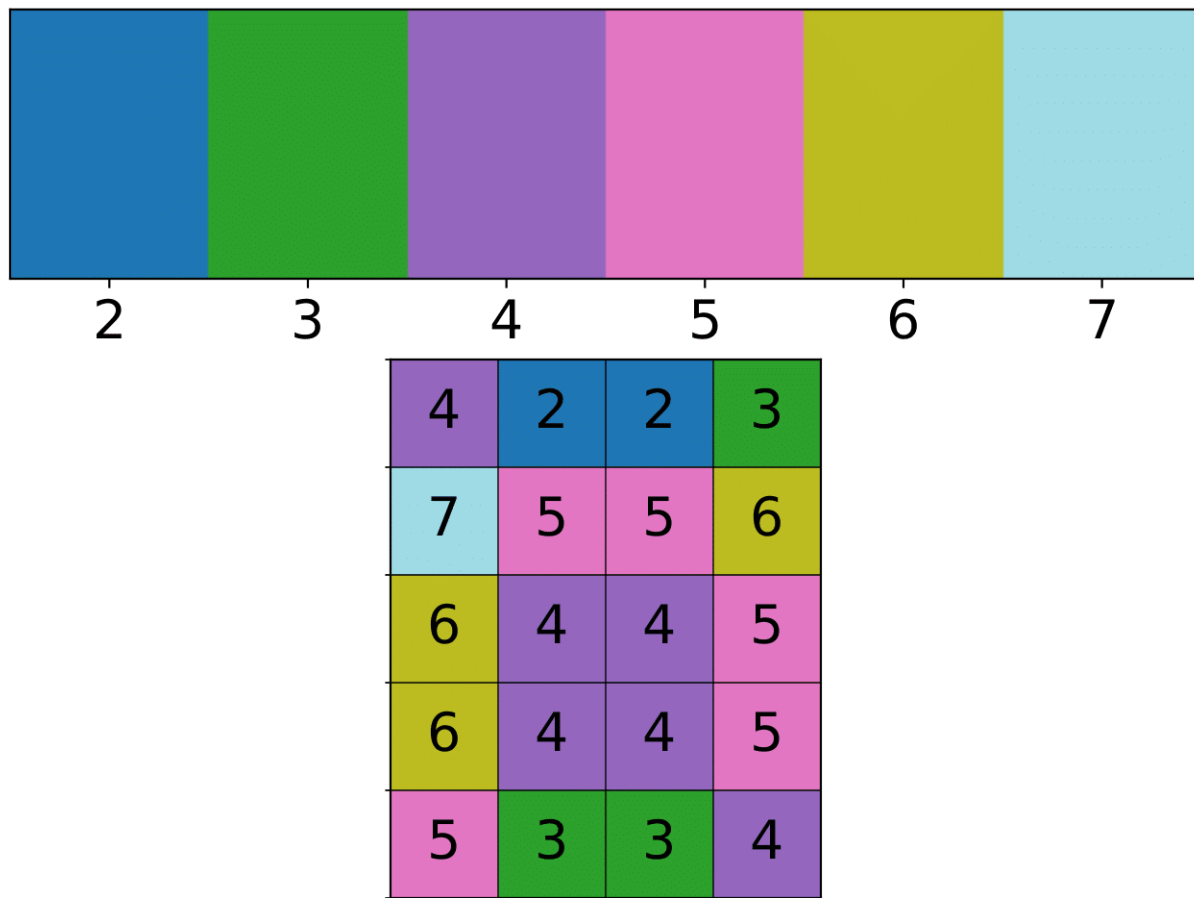
У цій задачі є умовні блоки. Якщо ваше рішення працює правильно для певних обмежень, воно отримає певну кількість балів. Зверніть увагу, що тестування відбувається тест за тестом, а не за блоками.

- (10 балів):  $m = 1$ ;
- (40 балів):  $n, m \leq 1\,000$ ;
- (50 балів): без додаткових обмежень.

### Приклад

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

### Зауваження



Візуалізація першого прикладу.

## Задача D. Процент виграшу

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: 1 second  
Ліміт використання пам'яті: 256 megabytes

Сакурако грає в шашки. Зараз, з  $q$  зіграних ігор, вона виграла  $p$  ігор ( $p < q$ ).

Вона хоче підвищити свій процент виграшу. Вона хоче, щоб він був не менше ніж  $\frac{a}{b}$  (де  $a$  та  $b$  — взаємно прості числа).

Цілі числа  $a$  та  $b$  називаються взаємно простими, якщо їхній НСД (найбільший спільний дільник) дорівнює одиниці.

Яка мінімальна кількість ігор, які їй потрібно зіграти, щоб її відсоток виграшу став не менше ніж  $\frac{a}{b}$  ( $a < b$ )?

Процент виграшу дорівнює  $\frac{x}{x+y}$ , де  $x$  — кількість ігор, які виграла Сакурако, а  $y$  — кількість ігор, які вона програла.

### Формат вхідних даних

Перший рядок містить два цілі числа  $p$  та  $q$  ( $0 \leq p < q \leq 10^6$ ).

Другий рядок містить два взаємно прості цілі числа  $a$  та  $b$  ( $0 \leq a < b \leq 10^6$ ).

### Формат вихідних даних

Виведіть мінімальну кількість ігор, які Сакурако потрібно зіграти, щоб її процент виграшу став не менше ніж  $\frac{a}{b}$ .

### Система оцінювання

У цій задачі є умовні блоки. Якщо ваше рішення працює правильно для певних обмежень, воно отримає певну кількість балів. Зверніть увагу, що тестування відбувається тест за тестом, а не за блоками.

- (34 бали):  $p, q \leq 10^4$ ;
- (66 балів): без додаткових обмежень.

### Приклад

standard input	standard output
1 3	5
8 11	

### Зауваження

Якщо Сакурако виграє всі п'ять ігор, її процент виграшу буде  $\frac{1+5}{3+5} = \frac{6}{8}$ , що більше ніж  $\frac{8}{11}$ . Можна показати, що якщо вона зіграє лише чотири гри, цього буде недостатньо.

## Задача Е. Картина Сакурако

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: 1 second  
Ліміт використання пам'яті: 256 megabytes

Класичні 2D картини нудні. Ось чому Сакурако вирішила намалювати картину, яка складається з  $10^{100}$  точок, що розташовані на одній лінії.

Спочатку вона хотіла намалювати деякі точки з  $n$  різними кольорами.

Для кожного кольору  $i$  вона вибрала відрізок точок, пронумерованих від  $l_i$  до  $r_i$ , які вона хотіла пофарбувати.

Після того, як ви це побачили, ви сказали, що жодна точка не повинна бути пофарбована більше ніж в **один** колір.

Оскільки Сакурако вперта, вона вирішила, що зробить точно одне з наступного для кожного кольору  $i$  ( $1 \leq i \leq n$ ):

- Повністю видалити колір  $i$  (отже, жодна точка не пофарбована в колір  $i$ ).
- Вибрати відрізок  $[x_i, y_i]$  так, щоб  $l_i \leq x_i \leq y_i \leq r_i$  і пофарбувати ці точки в колір  $i$ .

Сакурако намагається максимізувати загальну кількість пофарбованих точок, так щоб жодна точка не повинна бути пофарбована більше ніж в **один** колір.

Оскільки вона зайнята підготовкою, вона попросила вас визначити максимальну кількість точок, які можуть бути пофарбовані.

Точка  $q$  вважається пофарбованою в колір  $i$ , якщо  $x_i \leq q \leq y_i$ .

### Формат вхідних даних

Перший рядок містить одне ціле число  $n$  ( $1 \leq n \leq 10^5$ ) — кількість кольорів, які спочатку вибрала Сакурако.

Кожен з наступних  $n$  рядків містить два цілі числа  $l_i$  та  $r_i$  ( $0 \leq l_i \leq r_i \leq 10^{15}$ ).

### Формат вихідних даних

Виведіть максимальну кількість точок, які можуть бути пофарбовані.

### Система оцінювання

У цій задачі є умовні блоки. Якщо ваше рішення працює правильно для певних обмежень, воно отримає певну кількість балів. Зверніть увагу, що тестування відбувається тест за тестом, а не за блоками.

1. (36 балів):  $l_i, r_i \leq 10^7$ ;
2. (64 бали): без додаткових обмежень.

### Приклад

standard input	standard output
5	7
1 1	
4 4	
5 8	
3 4	
4 5	

### Зауваження

Один з варіантів досягнення семи точок:

1. Перший колір. Виберіть  $x = 1$  та  $y = 1$ .

2. Другий колір. Виберіть  $x = 4$  та  $y = 4$ .
3. Третій колір. Виберіть  $x = 5$  та  $y = 8$ .
4. Четвертий колір. Виберіть  $x = 3$  та  $y = 3$ .
5. П'ятий колір. Видаліть його.

## Задача F. Кольорові кола

Назва вхідного файлу: `standard input`  
Назва вихідного файлу: `standard output`  
Ліміт часу: 1 second  
Ліміт використання пам'яті: 256 megabytes

Сакурако знайшла  $n$  кіл, розташованих на прямій лінії.  $i$ -те коло розташоване на координаті  $x_i$  з радіусом  $r_i$ . Спочатку кожне коло поділено на дві частини: ліва половина є фіолетовою, а права половина є жовтою.

Сакурако дозволено обертати деякі з кіл, що призведе до зміни їх кольорів — ліва половина стане жовтою, а права половина стане фіолетовою.

Допоможіть Сакурако визначити, чи можливо обернути кола так, щоб жодні два кола не мали перекриваючихся ділянок різних кольорів. Якщо це можливо, надайте необхідні обертання; в іншому випадку, визначте, що це неможливо.

### Формат вхідних даних

Перший рядок містить одне ціле число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — кількість кіл.

Другий рядок містить  $n$  цілих чисел  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq 10^9$ ) — координати кіл.

Третій рядок містить  $n$  цілих чисел  $r_1, r_2, \dots, r_n$  ( $1 \leq r_i \leq 10^9$ ) — радіуси кіл.

### Формат вихідних даних

Виведіть  $-1$ , якщо неможливо обернути кола, як описано в умові. Якщо це можливо, виведіть рядок довжини  $n$ , де  $i$ -ий символ є "0", якщо Сакурако не обертає  $i$ -те коло, і "1", якщо обертає.

**Якщо є кілька відповідей, ви можете вивести будь-яку.**

### Система оцінювання

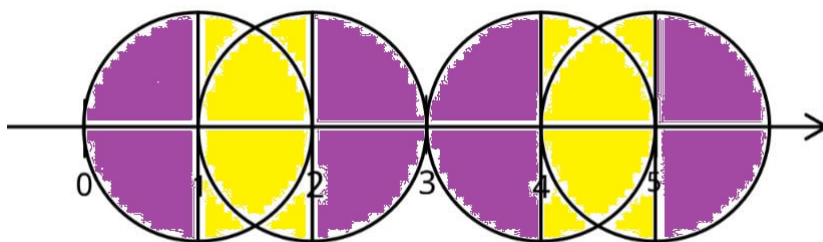
У цій задачі є умовні блоки. Якщо ваше рішення працює правильно для певних обмежень, воно отримає певну кількість балів. Зверніть увагу, що тестування відбувається тест за тестом, а не за блоками.

- (40 балів):  $r_1 = r_2 = \dots = r_n$ ;
- (60 балів): без додаткових обмежень.

### Приклади

standard input	standard output
4 1 2 4 5 1 1 1 1	0101
3 1 2 3 2 2 2	-1

### Зауваження



Візуалізація першого прикладу.



## Задача G. Словник префіксів і суфіксів

Назва вхідного файлу:	standard input
Назва вихідного файлу:	standard output
Ліміт часу:	1 second
Ліміт використання пам'яті:	256 megabytes

Сакурако тільки що знайшла найстаріший словник у всесвіті. Він містить більше ніж  $n$  слів. Сакурако може створювати нові слова на основі цих слів.

Кожне нове слово може бути сконструйоване шляхом взяття префікса деякого слова зі словника та конкатенації (з'єднання) його з суфіксом деякого слова (може бути те саме слово). Довжина новоствореного слова повинна бути не більше  $10^3$ . Можна вибирати порожні префікси та суфікси.

Кожного разу Сакурако може створити нове слово, вибираючи будь-які два існуючі слова (навіть ті, які вона створила).

Сакурако цікавиться, чи можливо створити її улюблене слово  $s$ , не створюючи більше ніж 2 000 нових слів.

### Формат вхідних даних

Перший рядок містить один рядок  $s$  ( $1 \leq |s| \leq 10^3$ ) — слово, яке цікавить Сакурако. Рядок містить лише малі літери англійського алфавіту.

Другий рядок містить одне ціле число  $n$  ( $1 \leq n \leq 10^5$ ) — кількість слів у словнику спочатку.

Кожен з наступних  $n$  рядків містить один рядок  $a_i$  ( $1 \leq |a_i| \leq 10^3$ ) — перші слова, які були в словнику, що містять лише малі літери англійського алфавіту. Гарантовано, що сума всіх довжин не перевищує  $2 \cdot 10^5$ .

### Формат вихідних даних

Виведіть "No" якщо це неможливо, і "Yes" в іншому випадку.

Якщо це можливо, виведіть ціле число  $k$  ( $0 \leq k \leq 2000$ ), кількість операцій, які потрібно виконати, щоб отримати рядок  $s$ , а наступні  $k$  рядків описують операції. Зверніть увагу, що нема потреби мінімізувати це значення.

Кожен рядок міститиме чотири цілі числа:

- $i_p$  ( $1 \leq i_p \leq t$ , де  $t$  — кількість рядків на той момент) — індекс рядка, з якого ми беремо префікс;
- $l_p$  ( $0 \leq l_p \leq |a_{i_p}|$ , де  $|a_{i_p}|$  — довжина  $i_p$ -го рядка) — довжина префікса, який ми хочемо взяти;
- $i_s$  ( $1 \leq i_s \leq t$ , де  $t$  — кількість рядків на той момент) — індекс рядка, з якого ми беремо суфікс;
- $l_s$  ( $0 \leq l_s \leq |a_{i_s}|$ ,  $1 \leq l_p + l_s \leq 10^3$ , де  $|a_{i_s}|$  — довжина  $i_s$ -го рядка) — довжина суфікса, який ми хочемо взяти.

$i$ -те слово матиме індекс  $n + i$ .

### Система оцінювання

У цій задачі є умовні блоки. Якщо ваше рішення працює правильно для певних обмежень, воно отримає певну кількість балів. Зверніть увагу, що тестування відбувається тест за тестом, а не за блоками.

1. (24 бали):  $s$  та  $a_i$  для всіх  $i$  містять лише літери  $a$  та  $b$ ;
2. (30 балів):  $n \leq 1000$ ;
3. (46 балів): без додаткових обмежень.

## Приклади

standard input	standard output
a 1 bac	Yes 3 1 2 1 0 2 0 2 1 3 1 3 0
ab 1 bcad	Yes 6 1 1 1 0 2 0 2 1 1 3 1 0 4 0 4 1 5 1 5 0 6 1 3 1