

## Задача А. Заявки

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

Закінчивши університет, Антон вирішив почати шукати роботу.

Відразу ж він зайшов на сайт **ConnectedIn**. Перейшовши у вкладку «Вакансії», йому видало дуже багато вакансій. Проте, на сайті не вказано точну кількість. Він знає, що на кожній сторінці максимум 20 вакансій, а вакансії займають  $n$  сторінок.

Ви гарно розібрались як працює сайт. Він намагається додавати вакансії на створенні сторінки, а якщо не вдалось додати, то створює нову сторінку. Іншими словами, коли додається нова вакансія — ця вакансія додається до останньої сторінки. Якщо ж на останній сторінці до додавання уже 20 вакансій — то створюється нова сторінка та вакансія додається уже туди.

Знаючи цю інформацію, скажіть мінімальну і максимальну кількість вакансій.

### Формат вхідних даних

Перший рядок вхідних даних містить одне ціле число  $n$  ( $1 \leq n \leq 100$ ) — кількість сторінок з вакансіями.

### Формат вихідних даних

Виведіть два цілі числа — мінімальну та максимальну кількість вакансій, яка могла бути на сайті.

### Приклади

standard input	standard output
3	41 60
14	261 280

## Задача В. Хрестики-нулики

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

Ви грали в хрестики нулики з другом онлайн. Однак сталася халепка, знову повітряна тривога у вашому районі! Ви, як вихована людина, йдете у бомбосховище. По закінченню тривоги Ви приходите догравати гру, але щось тут не так. Ваш друг міг обманути й змінити поле.

Ви запам'ятали поле гри  $A$  коли йшли. Повернувшись, ви бачите ту саму гру і поле  $B$ . Скажіть, чи можливо отримати з поля  $A$  поле  $B$  за не більше ніж один крок, зроблений за правилами?

Нагадаємо, що перший крок робить гравець, що ставить **хрестики**. Зверніть увагу, що тут не діє стандартне правило про те, що якщо є три хрестики чи нулика підряд, то гра зупиняється.

### Формат вхідних даних

Перші три рядки містять по три символи  $A_{i,j}$  — опис початкового поля.

Наступні три рядки містять по три символи  $B_{i,j}$  — опис кінцевого поля.

Кожна клітинка поля описується трьома символами:

- «.» — позначає пусту клітинку;
- «0» — позначає поставлений нуль;
- «X» — позначає поставлений хрестик.

Гарантується, що поле  $A$  можна отримати послідовністю правильних ходів з пустого поля.

### Формат вихідних даних

Виведіть «YES» чи «NO» (у будь-якому регістрі) залежно від того, чи можна було отримати поле  $B$  з поля  $A$ .

## Приклади

standard input	standard output
.X. .X. 00. .X. .X. 000	NO
.XX .00 ... XXX .00 ...	YES
XXX 000 ... XXX 000 .X.	YES
OXO X.X OXO XOX O.O XOX	NO
.X. ... ... .X. ... ...	YES

## Примітка

У першому прикладі додається один нулик, проте зараз черга поставити хрестик.

У другому прикладі додається один хрестик.

У третьому прикладі також додається один хрест. Зверніть увагу, що попри те, що є три нулики (і хрестики) підряд — гра не зупиняється.

У четвертому прикладі поле змінене.

У п'ятому прикладі жодних кроків не було зроблено.

## Задача С. Дискотека

Обмеження часу: 3 seconds  
Обмеження використання пам'яті: 256 megabytes

Антон, як вчитель в школі, організовує дискотеку для дітей. Йому було доручено вибрати музикальний плейлист такий, щоб всі були в захваті від свята. Але в Антона величезна база пісень, а час дискотеки, на жаль, обмежений :(.

А саме, у базі є  $n$  пісень. Кожну пісню можна схарактеризувати двома цілими числами  $t_i$  і  $r_i$  — довжина пісні, та її рейтинг. Антон, як меломан, хоче вибрати **рівно**  $k$  пісень (тобто, не можна вибрати менше), щоб максимізувати відношення суми рейтингу до суми довжин.

Більш формально, нехай  $S$  — множина пісень і  $X \subseteq S$ ,  $|X| = k$  — підмножина пісень, яка була вибрана в плейлист. Треба максимізувати  $\frac{\sum_{i \in X} r_i}{\sum_{i \in X} t_i}$ . Тобто, потрібно знайти суму чисел  $r_i$  всіх пісень, які будуть грати на дискотеці, знайти суму чисел  $t_i$  всіх пісень, які будуть грати на дискотеці, та поділити перше на друге. Ось це число потрібно максимізувати.

Знайдіть та виведіть максимально можливе відношення.

### Формат вхідних даних

Перший рядок містить два цілі числа  $n, k$  ( $1 \leq k \leq n \leq 10^5$ ) — сумарна кількість пісень та кількість пісень, які мають бути вибрані в плейлист.

Другий рядок містить  $n$  цілих чисел  $r_i$  ( $1 \leq r_i \leq 10^5$ ).

Третій рядок містить  $n$  цілих чисел  $t_i$  ( $1 \leq t_i \leq 10^5$ ).

### Формат вихідних даних

Виведіть одне число — максимальне відношення.

Ваша відповідь буде вважатися правильною, якщо його абсолютна або відносна похибка не перевищує  $10^{-4}$ .

Формально, нехай ваша відповідь рівна  $a$ , а відповідь журі рівна  $b$ . Ваша відповідь буде зарахована, якщо і лише якщо  $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-4}$ .

### Приклади

standard input	standard output
2 2 1 2 2 3	0.600000
5 2 1 2 3 2 3 2 4 2 5 3	1.200000

### Примітка

У першому прикладі є дві пісні, які й потрібно додати в плейлист. Відповідно, відношення буде таким:

$$\frac{1+2}{2+3} = \frac{3}{5} = 0.6$$

У другому прикладі найкраще взяти третю та п'яту пісню. Відповідно, відношення буде таким:

$$\frac{3+3}{2+3} = \frac{6}{5} = 1.2$$

## Задача D. Послідовність

Обмеження часу: 1.5 seconds  
Обмеження використання пам'яті: 512 megabytes

На Антона тиснуть терміни — треба здавати всі роботи. Як часто це буває — продовжити дедлайн він не може...

Вам дана послідовність  $a$  з  $n$  цілих чисел, і два цілі числа  $l$  та  $r$ . Вам треба знайти найдовшу таку підпослідовність  $b$  послідовності  $a$ , що  $l \leq b_i + b_{i+1} \leq r$  ( $1 \leq i < |b|$ ). Тут  $|b|$  позначає кількість елементів послідовності  $b$ . Іншими словами, вам потрібно вибрати таку підпослідовність, що сума кожних двох сусідніх чисел не менша за  $l$  та не більша за  $r$ .

Підпослідовність масиву — це послідовність, яку можна отримати видаленням кількох (можливо жодного) елементів початкової послідовності.

### Формат вхідних даних

Перший рядок містить три цілі числа  $n, l, r$  ( $1 \leq n \leq 5 \cdot 10^5, 1 \leq l \leq r \leq 10^{17}$ ).  
Другий рядок містить  $n$  цілих чисел  $a_i$  ( $1 \leq a_i \leq r$ ) — опис послідовності.

### Формат вихідних даних

Виведіть єдине ціле число — максимальну довжину такої підпослідовності  $b$ .

### Система оцінювання

- (1 бал): всі  $a_i$  однакові;
- (3 бали):  $a_i = a_{i+2}$  для всіх  $1 \leq i \leq n - 2$ ;
- (9 балів):  $n \leq 20$ ;
- (8 балів):  $n \leq 5000$ ;
- (9 балів):  $r - l \leq 10$ ;
- (10 балів):  $l = 1, r \leq 10^6$ ;
- (13 балів):  $r \leq 10^6$ ;
- (10 балів):  $l = 1$ ;
- (24 бали):  $n \leq 10^5$ ;
- (13 балів): без додаткових обмежень.

### Приклади

standard input	standard output
5 2 6 1 3 4 2 5	3
2 1 1 1 1	1

### Примітка

У першому прикладі можна вибрати таку підпослідовність  $[1, 3, 2]$ .  $2 \leq 1 + 3 \leq 6$ .  $2 \leq 3 + 2 \leq 6$ .  
Також можна вибрати й  $[1, 4, 2]$ .

## Задача Е. Матчі

Обмеження часу: 2 seconds  
Обмеження використання пам'яті: 256 megabytes

Антон винайшов новий командний вид спорту — вотербол (як пейнтбол, тільки з водою). Він захотів поділитись своїм винаходом і покликав  $n$  друзів. В Антона прекрасні відносини з усіма друзями, проте, не факт, що між собою друзі мають такі ж. А саме, ми знаємо, що друг номер  $a_i$  конфліктує з другом  $b_i$ .

Антону дали записані  $m$  конфліктних пар  $(a_i; b_i)$ . І от наче, можна було б поділити гравців на дві команди, але так просто в Антона не буває...

Він хоче взяти розбити ці  $m$  конфліктні пари на відрізки так, щоб

1. кожна конфліктна пара належала рівно до одного відрізка;
2. якщо враховувати тільки відносини на кожному відрізку окремо, то можна розбити всіх людей на дві команди так, щоб не було двох людей, що конфліктують між собою, та знаходяться в одній команді.

Наприклад, нехай у нас є масив конфліктних пар  $[(1, 2), (2, 3), (1, 3)]$ . Ми можемо взяти перші дві пари у перший відрізок. У такому випадку ми зможемо зробити команди:  $[1, 3]$  та  $[2]$ . У другий відрізок ми можемо взяти останню пару. 1 та 3 мають бути в різних команда, а 2 може бути у будь-якій команді. Альтернативно, ми можемо віднести першу пару до першого відрізка, а останні два до другого. Зверніть увагу, що ми не можемо віднести першу та третю пару до одного відрізка, а другу пару до іншого. Причиною цього є те, що відрізок має містити лише послідовні пари. Ми також не можемо віднести всі пари до одного відрізка, бо тоді у будь-якому випадку буде команда, у якій люди конфліктують.

Антон знову перемудрив з умовами, і тепер не може розв'язати задачу. Допоможіть йому, і скажіть мінімальну кількість відрізків, на які він може розбити пари, щоб виконувались умови вище.

### Формат вхідних даних

Перший рядок містить два цілі числа  $n, m$  ( $1 \leq n, m \leq 10^6$ ) — кількість друзів та недружніх відносин серед друзів.

Наступні  $m$  рядків містять по два цілі числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ), що означають, що  $a_i$ -й друг конфліктує з  $b_i$ -м.

Гарантується, що жодна пара  $(a; b)$  не повторюється більше ніж один раз.

### Формат вихідних даних

Виведіть одне ціле число — відповідь на задачу.

### Система оцінювання

1. (4 бали):  $n \leq 3$ ;
2. (7 балів):  $n \leq 10$ ;
3. (15 балів):  $n, m \leq 5000$ ;
4. (13 балів): пари конфліктуючих друзів у вхідних даних згенеровано випадково; це значить, що було вибрано випадковим чином  $m$  пар з усіх  $\frac{n(n-1)}{2}$  пар;
5. (14 балів): кожна людина конфліктує не більше, ніж з 10 людьми;
6. (19 балів):  $n \leq 10^5$ ;
7. (17 балів):  $n \leq 2 \cdot 10^5$ ;
8. (11 балів): без додаткових обмежень.

## Приклади

standard input	standard output
3 3 1 2 2 3 1 3	2
5 10 2 4 1 2 3 4 1 3 1 5 4 5 2 3 3 5 1 4 2 5	3

## Примітка

Перший приклад пояснений в легенді вище.

У другому прикладі, можна, наприклад, розбити на наступні відрізки: [1; 6], [7; 9], [10; 10].

На першому відрізку можна утворити команди [1, 4], [2, 3, 5] — 1 і 4 не конфліктують між собою, як і пари (2; 3), (2; 5), (3; 5).

На другому відрізку можна утворити команди [1, 3], [2, 4, 5] — 1 і 3 не конфліктують між собою, як і пари (2; 4), (2; 5), (4; 5).

На третьому відрізку можна утворити команди [1, 2], [3, 4, 5] — 1 і 2 не конфліктують між собою, як і пари (3; 4), (3; 5), (4; 5).