

## Note

Задачи D, F, H, I, J и K взяты с Codeforces. Мы благодарим проект Codeforces, Михаила Мирзаянова и авторов задач за возможность их использования на наших сборах.

## Задача А. Наибольшая возрастающая подпоследовательность

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана последовательность, требуется найти её наибольшую возрастающую подпоследовательность.

### Формат входного файла

В первой строке входных данных задано целое число  $n$  — длина последовательности ( $1 \leq n \leq 300\,000$ ). Во второй строке задается сама последовательность. Числа разделяются пробелом. Элементы последовательности — целые числа, не превосходящие  $10^9$  по модулю.

### Формат выходного файла

В первой строке выведите длину наибольшей возрастающей подпоследовательности, а во второй строке выведите через пробел саму наибольшую возрастающую подпоследовательность данной последовательности. Если ответов несколько — выведите любой.

### Пример

<code>stdin</code>	<code>stdout</code>
6	3
3 29 5 5 28 6	3 5 28

## Задача В. Число возрастающих подпоследовательностей

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Задана последовательность из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Подпоследовательностью длины  $k$  этой последовательности называется набор индексов  $i_1, i_2, \dots, i_k$ , удовлетворяющий неравенствам  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Подпоследовательность называется *возрастающей*, если выполняются неравенства  $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ .

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности  $a_1, \dots, a_n$ . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на  $10^9 + 7$ .

### Формат входного файла

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 100000$ ). Вторая строка входного файла содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$ . Все  $a_i$  не превосходят  $10^9$  по абсолютной величине.

### Формат выходного файла

В выходной файл выведите ответ на задачу.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

## Задача С. Наибольшая общая подпоследовательность

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

### Формат входного файла

В первой строке входных данных содержится целое число  $N$  — длина первой последовательности ( $1 \leq N \leq 2000$ ). Во второй строке заданы члены первой последовательности (через пробел) — целые числа, не превосходящие  $10^9$  по модулю. В третьей строке записано целое число  $M$  — длина второй последовательности ( $1 \leq M \leq 2000$ ). В четвертой строке задаются члены второй последовательности (через пробел) — целые числа, не превосходящие  $10^9$  по модулю.

### Формат выходного файла

В первой строке выведите длину наибольшей общей подпоследовательности, а во второй строке выведите через пробел саму наибольшую общую подпоследовательность данных последовательностей. Если ответов несколько — выведите любой.

### Пример

<code>stdin</code>	<code>stdout</code>
3	2
1 2 3	2 3
4	
2 3 1 5	

## Задача D. Конфеты и Камни

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 20 seconds  
Ограничение по памяти: 45 megabytes

Мальчик Геральд и его тренер Миша играют в интересную игру. В начале игры имеются куча из  $n$  конфет и куча из  $m$  камней. Геральд и Миша ходят по очереди, первым ходит Миша. Миша на своем ходу проверяет, сколько на данный момент Геральд съел конфет и камней. Пусть Геральд съел  $a$  конфет и  $b$  камней. Тогда Миша начисляет Геральду  $f(a, b)$  призовых очков. Геральд же на своем ходу съедает либо одну конфету из кучи с конфетами, либо один камень из кучи с камнями. Когда Миша обнаруживает, что Геральд съел все, кроме одной конфеты и одного камня, он последний раз начисляет очки и игра заканчивается. Опустошать ни ту, ни другую кучку Геральд не имеет права. Расскажите Геральду, как ему играть, чтобы получить наибольшее количество очков: требуется найти один из возможных оптимальных вариантов игры для Геральда.

### Формат входного файла

В первой строке содержатся три целых числа  $n, m, p$  ( $1 \leq n, m \leq 20000$ ,  $1 \leq p \leq 10^9$ ). Во второй строке находятся  $n$  целых чисел  $x_0, x_1, \dots, x_{n-1}$  ( $0 \leq x_i \leq 20000$ ). В третьей строке находятся  $m$  целых чисел  $y_0, y_1, \dots, y_{m-1}$  ( $0 \leq y_i \leq 20000$ ). Величина  $f(a, b)$  вычисляется, как остаток от деления суммы  $x_a + y_b$  на число  $p$ .

### Формат выходного файла

В первой строке выведите единственное число: максимальное количество призовых очков, которое может заработать Геральд. Во второй строке выведите строку из  $n + m - 2$  символов, каждый из которых — это «C» или «S»,  $i$ -ый символ должен быть «C», если на своем  $i$ -ом ходу Геральд должен съесть конфету, и «S», если камень.

### Примеры

stdin	stdout
2 2 10 0 0 0 1	2 SC
3 3 10 0 2 0 0 0 2	10 CSSC
3 3 2 0 1 1 1 1 0	4 SCSC

### Note

В первом тесте, если на первом ходу Геральд съест камень, то после него он получит одно очко, а если конфету — то ноль. Перед первым своим ходом Геральд получит в любом случае 0 очков, а после второго — в любом случае 1. Таким образом, максимальное количество очков, которое может получить Геральд равно 2, и для этого надо съесть сначала камень, потом конфету.

## Задача Е. Наибольшая общая возрастающая подпоследовательность

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны две последовательности чисел —  $a$  и  $b$ . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие  $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$  и  $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$ , что  $\forall t : a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$  и  $k$  максимально.

### Формат входного файла

На первой строке целые числа  $n$  и  $m$  от 1 до 3 000 — длины последовательностей. Вторая строка содержит  $n$  целых чисел, задающих первую последовательность. Третья строка содержит  $m$  целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие  $10^9$ .

### Формат выходного файла

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

### Примеры

<code>stdin</code>	<code>stdout</code>
6 5 1 2 1 2 1 3 2 1 3 2 1	2

## Задача F. И снова НОП!

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам дана строка  $S$  длины  $n$ , где каждый символ является одной из первых  $m$  строчных букв английского алфавита.

Подсчитайте, сколько различных строк  $T$  длины  $n$ , состоящих из первых  $m$  букв английского алфавита существует, таких, что длина НОП (наибольшей общей подпоследовательности)  $S$  и  $T$  равняется  $n - 1$ .

Напомним, что НОП двух строк,  $S$  и  $T$  — это наибольшая строка  $C$ , такая, что  $C$  встречается как в  $S$ , так и в  $T$  как подпоследовательность.

### Формат входного файла

В первой строке записано два числа,  $n$  и  $m$ , обозначающих длину строки  $S$  и количество первых строчных букв английского алфавита, из которых формируется набор символов для строк ( $1 \leq n \leq 100\,000$ ,  $2 \leq m \leq 26$ ).

Во второй строке записана сама строка  $S$ .

### Формат выходного файла

Выведите ответ в единственной строке.

### Примеры

стандартный ввод	стандартный вывод
3 3 aaa	6
3 3 aab	11
1 2 a	1
10 9 abacdefgh	789

### Note

В первом примере 6 возможных строк  $T$  таковы: aab, aac, aba, аса, баа, саа.

Во втором примере 11 возможных строк  $T$  таковы: aaa, aac, aba, abb, abc, аса, acb, баа, bab, саа, sab.

В третьем примере единственная возможная строка  $T$  такова: b.

## Задача G. Покупка участков

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 0.5 секунды  
Ограничение по памяти: 256 мегабайт

Фермер Паша собирается купить как можно больше земли для своей фермы. Он присмотрел  $n$  ( $1 \leq n \leq 50\,000$ ) прямоугольных участков,  $i$ -й из которых имеет размер  $w_i \times h_i$ , где  $1 \leq w_i, h_i \leq 10^6$ .

Если фермер Паша покупает участок земли, он платит один доллар за каждую единицу площади участка. Но если он покупает сразу несколько участков, то действует система скидок. А именно, он платит  $\max w \cdot \max h$  за множество участков, где максимум берется по всем покупаемым участкам. Например, если фермер Паша покупает участки  $5 \times 3$  и  $3 \times 5$ , он заплатит  $5 \times 5 = 25$  долларов.

Фермер Паша хочет купить все участки и потратить минимальную сумму. Помогите ему выбрать, как разбить участки на наборы, чтобы купить их и заплатить суммарно минимальную сумму.

### Формат входного файла

Первая строка содержит число  $n$ . Следующие  $n$  строк содержат по два целых числа: размеры участков.

### Формат выходного файла

Выведите одно число: минимальную сумму, которую сможет потратить фермер Паша.

### Примеры

<code>stdin</code>	<code>stdout</code>
4 100 1 15 15 20 5 1 100	500



## Задача Н. Калила и Димна на лесозаготовках

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Калила и Димна — два шакала. Они живут в огромных джунглях. Однажды шакалы решили устроиться на завод лесозаготовки и подработать.

Управляющий завода хочет, чтобы они отправились в джунгли и срубили  $n$  деревьев высотой  $a_1, a_2, \dots, a_n$ . Для этого Калила и Димна купили цепную пилу в магазине. Каждый раз, когда они используют пилу на дереве номер  $i$ , они уменьшают высоту этого дерева на единицу. Каждый раз Калила и Димна должны заправить пилу для использования. Цена заправки зависит от того, какие деревья полностью спилены (дерево считается полностью спиленным, если его высота равна 0). Если максимальный идентификатор полностью срубленного дерева равняется  $i$  (первоначально это дерево имело высоту  $a_i$ ), то цена заправки пилы равняется  $b_i$ . Если ни одно дерево не срублено полностью, то заправлять пилу запрещается. Изначально пила заправлена. Известно, что для каждого  $i < j$ ,  $a_i < a_j$  и  $b_i > b_j$ , а также  $b_n = 0$  и  $a_1 = 1$ .

Калила и Димна хотят полностью срубить все деревья с минимальными затратами. Они ждут Вашей помощи! Поможете?

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10^5$ ). Во второй строке записано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ). В третьей строке записано  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^9$ ).

Гарантируется, что  $a_1 = 1$ ,  $b_n = 0$ ,  $a_1 < a_2 < \dots < a_n$  и  $b_1 > b_2 > \dots > b_n$ .

### Формат выходного файла

В единственной строке должна быть записана минимальная стоимость вырубания всех деревьев.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на C++. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

### Примеры

stdin	stdout
5 1 2 3 4 5 5 4 3 2 0	25
6 1 2 3 10 20 30 6 5 4 3 2 0	138

## Задача I. Транспортировка кошек

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Zxr960115 содержит большое хозяйство. Он кормит  $m$  милых кошечек и держит у себя  $p$  кормильщиков. Через ферму проходит прямая дорога, а вдоль дороги расположено  $n$  холмов, пронумерованных от 1 до  $n$ , слева направо. Расстояние от холма  $i$  до  $i - 1$  равняется  $d_i$  метров. Кормильщики живут на холме 1.

Однажды кошечкам захотелось порезвиться и они разбежались. Кошка  $i$  пошла к холму  $h_i$ , дошла до него в момент времени  $t_i$ , а затем стала ждать кормильщика на холме  $h_i$ . Кормильщики должны собрать всех разбежавшихся кошек. Каждый кормильщик идет прямо от холма номер 1 до холма номер  $n$ , не останавливаясь у какого-либо холма, и собирает всех кошек, **ожидающих** на каждом холме. Кормильщики двигаются со скоростью 1 в единицу времени и достаточно сильны, чтобы собрать сколько угодно кошек.

Например, пусть имеется два холма ( $d_2 = 1$ ) и одна кошечка, которая дошла до холма 2 ( $h_1 = 2$ ) в момент времени 3. Тогда, если кормильщик отправится за кошками от холма 1 в момент времени 2 или 3, то он сможет забрать эту кошку. Но если он отправится от холма 1 в момент времени 1, то он не сможет этого сделать. Если кормильщик отправится за кошкой в момент времени 2, то кошка будет ждать его 0 единиц времени, если же он отправится в момент времени 3, то кошка будет ждать его 1 единицу времени.

Ваша задача — составить расписание отправки от холма 1 для кормильщиков так, чтобы общее время ожидания кошек до того как их заберут было минимальным.

### Формат входного файла

В первой строке входных данных содержится три целых числа  $n, m, p$  ( $2 \leq n \leq 10^5, 1 \leq m \leq 10^5, 1 \leq p \leq 100$ ).

Во второй строке содержится  $n - 1$  положительных целых чисел  $d_2, d_3, \dots, d_n$  ( $1 \leq d_i < 10^4$ ).

В каждой из следующих  $m$  строк содержится по два целых числа  $h_i$  и  $t_i$  ( $1 \leq h_i \leq n, 0 \leq t_i \leq 10^9$ ).

### Формат выходного файла

Выведите целое число, минимальную сумму времен ожидания всех кошек.

Пожалуйста, не используйте спецификатор `%lld` для чтения и записи 64-битных чисел на C++. Рекомендуется использовать потоки `cin, cout` или спецификатор `%I64d`.

### Примеры

<code>stdin</code>	<code>stdout</code>
4 6 2 1 3 5 1 0 2 1 4 9 1 10 2 10 3 12	3

## Задача J. Ciel и гондолы

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Лиса Ciel зашла в парк аттракционов. И вот, она в очереди на колесо обозрения. В очереди стоит  $n$  людей (хотя нет, скорее лис): мы будем считать, что первая лиса стоит в начале очереди, а  $n$ -ая лиса стоит в хвосте очереди.

Всего имеется  $k$  гондол, мы распределяем лис по гондолам следующим образом:

- Когда подплывает первая гондола,  $q_1$  лис переходят из начала очереди в подплывшую гондолу.
- Затем, когда подплывает вторая гондола,  $q_2$  лис из начала оставшейся очереди переходит в эту гондолу.
- ...
- Оставшиеся  $q_k$  лис идут с последнюю ( $k$ -ую) гондолу.

Обратите внимание, что числа  $q_1, q_2, \dots, q_k$  должны быть положительными. Из условия следует, что  $\sum_{i=1}^k q_i = n$  и  $q_i > 0$ .

Вы знаете как лисам не хочется задерживаться в гондолах с незнакомцами. Итак, Ваша задача — найти оптимальный способ размещения (то есть определить оптимальную последовательность  $q$ ), чтобы угодить всем. Для каждой пары лис  $i$  и  $j$  задано значение  $u_{ij}$ , обозначающее степень незнакомости. Можете считать, что  $u_{ij} = u_{ji}$  для всех  $i, j$  ( $1 \leq i, j \leq n$ ) и что  $u_{ii} = 0$  для всех  $i$  ( $1 \leq i \leq n$ ). Тогда значение незнакомости в гондоле определяется как сумма значений незнакомости между всеми парами лис, которые находятся в этой гондоле. Общее значение незнакомости определяется как сумма значений незнакомости по всем гондолам.

Помогите лисе Ciel найти минимальное возможное значение общей незнакомости при некотором оптимальном распределении лис по гондолам.

### Формат входного файла

В первой строке даны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 4000$  and  $1 \leq k \leq \min(n, 800)$ ) — количество лис в очереди и количество гондол. В следующих  $n$  строках записано по  $n$  целых чисел — матрица  $u$ , ( $0 \leq u_{ij} \leq 9$ ,  $u_{ij} = u_{ji}$  и  $u_{ii} = 0$ ).

Пожалуйста, используйте методы быстрого чтения (например, для Java используйте `BufferedReader` вместо `Scanner`).

### Формат выходного файла

Выведите целое число — минимальное возможное значение общей незнакомости.

## Примеры

stdin	stdout
5 2 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0	0
8 3 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0	7
3 2 0 2 0 2 0 3 0 3 0	2

## Note

В первом примере можно распределить лис вот так: 1, 2 идут в одну гондолу, 3, 4, 5 идут в другую гондолу.

Во втором примере оптимальное распределение таково: 1, 2, 3 | 4, 5, 6 | 7, 8.

## Задача К. Групповые проекты

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

В аудитории есть  $n$  студентов, которые работают над групповыми проектами. Студенты будут разделены на группы (группа может состоять и из одного студента), после чего каждый студент в группе будет работать над своей независимой частью проекта, а затем они объединят результаты в рамках этой группы. Известно, что  $i$ -й студент выполнит свою работу ровно за  $a_i$  минут.

Если студенты в группе работают с разной скоростью, то это раздражает как тех, кто делает свою работу быстро и ждёт остальных, так и тех, кто делает её медленно и задерживает всю группу. *Несбалансированностью* группы называется разница между максимальным  $a_i$  в группе и минимальным  $a_i$  в группе. Обратите внимание, что несбалансированность группы из одного студента равна 0. Сколькими способами можно разбить студентов на группы так, чтобы суммарная несбалансированность всех групп не превосходила  $k$ ?

Два разбиения считаются различными, если какие-то два студента были в одной группе в одном разбиении, но оказались в разных группах в другом разбиении.

### Формат входного файла

В первой строке записаны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 200$ ,  $0 \leq k \leq 1000$ ) — количество студентов и максимальная разрешённая суммарная несбалансированность соответственно.

Во второй строке записаны  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 500$ ) —  $i$ -е число равняется количеству минут, которое тратит на выполнение своей независимой части работы студент номер  $i$ .

### Формат выходного файла

Выведите количество способов разбить студентов на группы требуемым образом. Поскольку ответ может быть достаточно большим, то выведите остаток от деления этого числа на  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 2 4 5	3
4 3 7 8 9 10	13
4 0 5 10 20 21	1

### Note

В первом примере возможны три разбиения:

- Первый и второй студенты образуют одну группу, а третий студент — другую. Итоговая несбалансированность равняется  $2 + 0 = 2$ .
- Первый студент будет в группе один, а второй и третий студенты образуют другую группу. Несбалансированность равняется  $0 + 1 = 1$ .
- Все три студента будут в разных группах. Несбалансированность будет равна 0.

В третьем примере суммарная несбалансированность должна быть равна 0, поэтому каждый студент должен работать индивидуально.