

## Задача А. Поврежденный пароль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

У Ви была очень важная секретная строка  $s$ . Он применил к ней следующую последовательность действий:

1. Выбрал позицию  $x$  в строке  $s$  ( $1 \leq x \leq |s|$ ) и поменял  $s_x$  на символ  $c$  ( $s_x \neq c$ ).
2. Несколько, возможно ноль, раз вставил в произвольное место строки произвольный символ. Символ можно вставлять перед первым символом, между двумя соседними, или после последнего.

В результате у него получилась строка  $t$ . Вам даны строки  $s$  и  $t$ . Помогите Ви вспомнить, какую он выбрал позицию  $x$  и символ  $c$ .

### Формат входных данных

В первой строке дана строка  $s$ , состоящая из строчных английских букв ( $1 \leq |s| \leq 10^6$ ).  
Во второй строке дана строка  $t$ , состоящая из строчных английских букв ( $|s| \leq |t| \leq 10^6$ ).  
Гарантируется, что  $t$  была получена из  $s$  в результате описанного процесса.

### Формат выходных данных

Выведите позицию  $x$  и символ  $c$ , которые Ви мог выбрать. Обратите внимание, что должно быть верно  $s_x \neq c$ . Если существует несколько ответов, вы можете вывести любой.

### Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	19	$ s ,  t  \leq 10$		первая ошибка
2	23	$ s ,  t  \leq 1\,000$	1	первая ошибка
3	21	$ s ,  t  \leq 5\,000$	1, 2	первая ошибка
4	18	$ s ,  t  \leq 100\,000$	1–3	первая ошибка
5	19	Без дополнительных ограничений	1–4	первая ошибка

### Пример

стандартный ввод	стандартный вывод
abc aacc	2 a

## Задача В. Пути файлов неисповедимы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Байтазар любит испытывать судьбу. Он ходит по лезвию бритвы, отправляет решения задач без тестирования на примерах, а также хочет, чтобы пути файлов имели максимально возможную длину, которую позволяет операционная система.

Когда Байтазар работает за чужим компьютером, может так случиться, что не все файлы удовлетворяют его требованию. В этом случае он создает символические ссылки и использует их для изменения длин. Ваша задача состоит в том, чтобы определить для каждого файла, возможно ли добавить одну символическую ссылку заданной длины  $s$  так, чтобы к этому файлу стало возможно обратиться по пути длины ровно  $k$  символов.

Если файл с именем `file` содержится в последовательности папок `dir1, dir2, ..., dirj`, то тогда абсолютный путь к этому файлу выглядит как `/dir1/dir2/.../dirj/file`. Корневой директории соответствует путь `/`, а каждому файлу, непосредственно содержащемуся в этой директории, соответствует абсолютный путь `/file`. Символическая ссылка — специальный файл, указывающий на одну из директорий, который может находиться внутри любой другой директории. В данной задаче запрещены символические ссылки на файлы. Используя символические ссылки, мы можем получить альтернативные пути файла. Например, если была создана символическая ссылка на корневую директорию `/` с именем `hello` в корневой директории `/`, то файловые пути `/dir/file`, `/hello/dir/file` и `/hello/hello/dir/file` будут указывать на один и тот же файл, однако будут иметь разную длину. Создав символическую ссылку на папку `/` с именем `hi` в директории `/dir`, мы получаем следующие альтернативные файловые пути: `/dir/file`, `/dir/hi/dir/file`, `/dir/hi/dir/hi/dir/file`. Обратите внимание, что символическая ссылка может указывать на любую директорию, включая те, которые расположены в дереве папок выше, ниже или в другом поддереве. Для упрощения задачи, `./`, `../` или `//` не используются для обращения к файлам.

### Формат входных данных

В первой строке входных данных содержатся числа  $n$ ,  $m$  и  $k$  — общее число папок (не включая корневую директорию), общее количество файлов и требуемую длину пути ( $1 \leq n, m \leq 3000$ ,  $1 \leq k \leq 1\,000\,000$ ). Корневая директория имеет номер 0, а все остальные папки пронумерованы от 1 до  $n$ . Файлы пронумерованы от 1 до  $m$ . Во второй строке входных данных содержится длина  $s$  символической ссылки, которую Байтазар может создать ( $1 \leq s \leq 1\,000\,000$ ). Имя ссылки в данной задаче не имеет значения: мы считаем, что всегда имеется возможность создать ссылку, которая не совпадает ни с одним другим именем. Следующие  $n$  строк описывают директории (не включая корневую). Каждое описание состоит из двух чисел  $p_i$  и  $l_i$ , которые обозначают, что название папки  $i$  имеет длину  $l_i$ , а родительская директория (та, в которой папка с номером  $i$  содержится непосредственно) имеет номер  $p_i$  ( $0 \leq p_i < i$ ,  $1 \leq l_i \leq 1\,000\,000$ ). Следующие  $m$  строк содержат описания файлов в том же формате: длина названия файла и номер родительской директории. Все файлы и папки имеют положительную длину, и никакой файл или папка не имеют длину пути больше  $k$  символов.

### Формат выходных данных

Для каждого файла выведите YES, если возможно создать ссылку длины  $s$  так, чтобы появился альтернативный путь длины  $k$  к этому файлу.

### Система оценки

Задача содержит три подзадачи.

1.  $n, m \leq 500$ . Стоимость подзадачи — 33 балла.
2.  $n, m \leq 3000$ , и для каждого файла, для которого ответ существует, можно найти также и ответ, в котором по ссылке нужно перемещаться не более одного раза. Стоимость подзадачи — 33 балла.

3. Нет дополнительных ограничений. Стоимость подзадачи — 34 балла.

### Пример

стандартный ввод	стандартный вывод
2 4 22	YES
2	YES
0 1	YES
1 5	NO
2 13	
2 10	
1 4	
0 7	

### Замечание

Разберем пример. Пусть символическая ссылка имеет имя LL, директории — a и bbbbb, а файлы — cccccccccc, dddddddd, eeee и ffffff. Корневая директория содержит папку a и файл ffffff; директория a содержит папку bbbbb и файл eeee; а папка bbbbb содержит файлы cccccccccc и dddddddd.

Путь к первому файлу /a/bbbbb/cccccccccc уже состоит из требуемого числа символов, таким образом, нам не нужна символическая ссылка. Для второго файла мы можем создать символическую ссылку /a/LL -> /a, тогда появится альтернативный путь /a/LL/bbbbb/ddddddddd. Для третьего файла мы можем создать символическую ссылку /a/LL -> /, а затем воспользоваться путем /a/LL/a/LL/a/LL/a/eeee. Для четвертого файла достичь требуемого невозможно.

## Задача С. Тройной запуск

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

На ближайшем всероссе планируется тестировать новый формат — тройной запуск программы. Однако в процессе подготовки задачи что-то усложнилось и теперь программа будет запускаться  $n$  раз. Всего существует 3 режима запуска программы, обозначим их буквами «X», «Y» и «N». Во время  $i$ -го запуска программы она будет запущена в режиме  $S_i$ .

К сожалению, PCMS не поддерживает много запусков программы, поэтому жюри решило сократить число запусков до нуля (задача будем иметь открытые тесты). Для этого жюри будет по одному удалять запуски программы, пока их не останется. Более формально, жюри выдаст перестановку  $p_0, p_1, \dots, p_{n-1}$  из чисел от 0 до  $n - 1$ , где  $p_i$  обозначает номер запуска в исходной нумерации, который удалится. Таким образом, в начале удалится запуск типа  $S_{p_1}$ , потом запуск  $S_{p_2}$ , потом запуск типа  $S_{p_3}$  и так далее, пока не удалится запуск типа  $S_{p_n}$ .

Жюри считает, что удаление запуска прошло хорошо, если удаляется запуск режима  $Y$ , и ближайший слева неудаленный запуск имеет типа  $X$ , а ближайший справа неудаленный запуск имеет тип  $N$ . Более формально, удаление проходит хорошо, если выполнены следующие условия:

- Сейчас удаляется запуск с номером  $b$  в исходной нумерации, и  $S_b = Y$ .
- Существует запуск с номером  $a$  в исходной нумерации, такой что он пока не удален,  $S_a = X$ , и все запуски с номерами больше  $a$  и меньше  $b$  уже удалены.
- Существует запуск с номером  $c$  в исходной нумерации, такой что он пока не удален,  $S_c = N$ , и все запуски с номерами больше  $b$  и меньше  $c$  уже удалены.

Во всех остальных случаях удаление не проходит хорошо и вредит задаче. Жюри намерены как можно меньше навредить задаче, поэтому они хотят максимизировать количество хороших удалений.

Вы наверное уже придумали как действовать жюри? Тогда держите усложнение задачи:

### Эта задача с двойным запуском

К сожалению, Жюри сами не знают, какие именно режимы планируются у всех  $n$  запусков. Это знает только разработчик задачи. А разработчик задачи пропал и связи с ним почти нет, да и сам он не особо хочет кому-то отвечать (такое иногда происходит с разработчиками задач для олимпиад). Чтобы не подводить жюри всеросса, разработчик задачи решил всё таки помочь им. Он может передать им некоторую строку  $t$  из нулей и единиц длины  $m$ . А дальше уже жюри по этой строке разработает последовательность удалений запусков программ, такую что количество хороший удалений будет максимально. Разработчик задачи хочет минимизировать своё общение со внешним миром, поэтому он хочет добиться того, чтобы  $m$  было как можно меньше. Помогите им разработать правильную стратегию.

### Формат входных данных

Ваша программа будет запускаться дважды. В первом запуске ваша программа будет работать за разработчика задачи, во втором — за жюри.

При первом запуске на первой строке ввода находятся два целых числа  $1$  и  $n$  ( $3 \leq n \leq 100\,000$ ) — номер запуска и длина последовательности  $s$ .

При первом запуске на второй строке ввода находится строка  $S$  — режимы запуска программ. Гарантируется, что строка  $S$  состоит только из букв «X», «Y» или «N».

При втором запуске на первой строке ввода находятся два целых числа  $2$  и  $n$  ( $3 \leq n \leq 100\,000$ ) — номер запуска и длина исходной последовательности  $s$ .

При втором запуске на второй строке ввода находится строка  $t$  из нулей и единиц. Гарантируется, что эта строка равна строке  $t$ , выведенной программой при первом запуске.

## Формат выходных данных

При первом запуске необходимо вывести строку  $t$ , состоящую из нулей и единиц — ту последовательность, которую передаст разработчик задачи жюри. Длина строки  $t$  не должна превышать 200 000.

При втором запуске по заданной строке  $t$  необходимо вывести перестановку из  $n$  чисел  $p_1, p_2, p_3, \dots, p_n$  ( $0 \leq p_i \leq n - 1$ ) — номера символов из последовательности  $S$  в исходной нумерации, в том порядке, в котором их надо удалять. Количество хороших удалений должно быть максимальным возможным.

**Обратите внимание, что вам необходимо сбрасывать буфер вывода.** Например, на языке C++ надо использовать функцию вызов `cout.flush()` или заканчивать вывод `cout << endl`, на Java вызов `System.out.flush()`, на Pascal `flush(output)` и `stdout.flush()` для языка Python.

## Система оценки

В данной задаче 2 группы тестов. Вы получите 0 баллов за любую группу тестов, в которой не было достигнуто максимальное возможное число хороших удалений.

В первой группе  $n \leq 18$ , группа оценивается в 15 баллов при прохождении всех тестов группы.

Во второй группе дополнительных ограничений нет. За группу вы можете получить не более 85 баллов. Обозначим за  $m$  максимальную длину последовательности  $t$  по всем тестам группы. Для вычисления балла используются следующие правила:

- При  $200\,000 < m$  ваш балл будет равен 0.
- При  $160\,000 < m \leq 200\,000$  ваш балл будет равен  $15 + \lfloor 10 \times \frac{200\,000 - m}{40\,000} \rfloor$
- При  $100\,000 < m \leq 160\,000$  ваш балл будет равен  $25 + \lfloor 30 \times \frac{160\,000 - m}{60\,000} \rfloor$
- При  $70\,000 < m \leq 100\,000$  ваш балл будет равен  $55 + \lfloor 30 \times (\frac{100\,000 - m}{30\,000})^2 \rfloor$
- При  $m \leq 70\,000$  ваш балл будет равен 85.

## Пример

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, если ваша программа выведет другую строку  $t$ , при втором запуске ввод также может быть другим.

Первый запуск.

стандартный ввод	стандартный вывод
1 4 XYXN	01

Второй запуск.

стандартный ввод	стандартный вывод
2 4 01	2 1 0 3

## Замечание

В тесте из условия исходная последовательность  $S$  равна XYXN.

Первым удаляется элемент с индексом 2, последовательность становится равна XY-N («-» обозначает удаленный элемент). Удаление не было хорошим.

Вторым удаляется элемент с индексом 1, последовательность становится равна X--N. Удаление было хорошим, так как удалился Y, предыдущим неудаленным элементом был X, в следующем неудаленным элементом был N.

Третим удаляется элемент с индексом 0, последовательность становится равна ---N. Удаление не было хорошим.

Четвертым удаляется элемент с индексом 3, последовательность становится равна ----. Удаление не было хорошим.

## Задача D. Вольеры для обезьян

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Как известно, все обезьяны (и в том числе Горилла Коко) — опасные животное. Они любят кидаться в людей гнилыми кокосами, разбрасываться фекалиями, громко кричать и причинять вред психике людей. Именно поэтому всех горилл обычно загоняют в вольеры, где им самое место.

Вольер для обезьян состоит из  $n$  секций, пронумерованных от 1 до  $n$ . В  $i$ -й секции содержится  $d_i$  обезьян. Известно, что для любого  $i$  от 1 до  $n - 1$  секция  $i$  отделена от секции  $i + 1$  перегородкой с номером  $i$ . Будем считать, что слева от 1 секции находится перегородка с номером 0, а справа от  $n$  секции находится перегородка с номером  $n$ . То есть для секции  $i$  слева находится перегородка  $i - 1$ , а справа — перегородка  $i$ .

Дрессировщик обезьян в течении  $q$  дней будет проводить по одному этапу дрессировки обезьян. Для каждого этапа дрессировки обезьян зафиксированы четыре числа  $x_i, y_i, l_i, r_i$ . Каждый этап дрессировки обезьян устроен следующим образом:

1. В начале дрессировщик заменяет число обезьян в секции  $x_i$  на  $y_i$ . То есть  $d_{x_i} = y_i$ . При этом во всех остальных секциях число обезьян остается прежним.
2. Берутся те секции обезьян, которые находятся между перегородкой  $l_i$  и  $r_i$  (то есть секции с номерами  $l_i + 1, l_i + 2, l_i + 3, \dots, r_i$ ).
3. Некоторые перегородки с номерами больше  $l_i$  но меньше  $r_i$  удаляются. Таким образом, некоторые секции объединяются. Дрессировщик хочет, чтобы число обезьян в получившихся секциях (между перегородками  $l_i$  и  $r_i$ ) образовывали чередующуюся последовательность максимальной длины. Определении чередующейся последовательности будет ниже.
4. В конце дрессировки перегородки возвращаются на места, число обезьян в каждой секции остается прежним (при этом изменение числа обезьян в секции  $x_i$  остается, то есть  $d_{x_i} = y_i$ ).

Последовательность чисел называется чередующейся, если каждое число либо меньше всех соседей, либо больше. Например последовательности  $(2, 9, 2, 7)$ ,  $(7, 1, 9, 4, 6)$ ,  $(5)$ ,  $(2, 1)$  чередующиеся, а последовательности  $(1, 2, 3)$ ,  $(7, 1, 4, 4, 6)$ ,  $(2, 2)$  не чередующиеся. Более формально, последовательность  $(a_1, a_2, \dots, a_m)$  чередующаяся, если хотя-бы одно из следующих условий выполнено:

- Для любого  $k$  от 1 до  $m - 1$  верно, что  $a_k < a_{k+1}$  для четных  $k$  и  $a_k > a_{k+1}$  для нечетных  $k$ .
- Для любого  $k$  от 1 до  $m - 1$  верно, что  $a_k > a_{k+1}$  для четных  $k$  и  $a_k < a_{k+1}$  для нечетных  $k$ .

К сожалению, так как дрессировщик слишком много времени проводит с обезьянами, он сильно отупел. Помогите ему для каждого дня узнать длину максимально длинной чередующейся последовательности в 3 пункте дрессировки.

### Формат входных данных

В первой строке дано одно целое число  $n$  ( $1 \leq n \leq 250\,000$ ) — число секций вольера.

Во второй строке дано  $n$  целых чисел  $d_1, d_2, d_3, \dots, d_n$  ( $1 \leq d_i \leq 10^9$ ) — число обезьян в каждой секции до первого дня дрессировки.

В третьей строке дано одно целое число  $q$  ( $q \leq 50\,000$ ) — число дрессировок.

В каждой из следующих  $q$  строк дано по четыре целых числа  $x_i, y_i, l_i, r_i$  ( $1 \leq x_i \leq n, 1 \leq y_i \leq 10^9, 0 \leq l_i < r_i \leq n$ ) — номер секции, в которой изменится число обезьян на  $i$ -й дрессировке, изменение числа обезьян в  $x_i$ -й секции, номера левой и правой перегородки, между которыми из обезьян будет строиться чередующаяся последовательность.

## Формат выходных данных

Выведите  $q$  строк. В  $i$ -й строке выведите максимальную по длине чередующуюся последовательность, которую дрессировщик сможет получить на  $i$ -й день.

## Система оценки

В данной задаче 6 групп тестов:

- (10 баллов)  $n \leq 200$ ,  $q \leq 10$
- (16 баллов)  $n \leq 2000$ ,  $q \leq 10$
- (14 баллов)  $q \leq 10$
- (26 баллов)  $y_i = d_{x_i}$  (то есть число обезьян в секциях не меняется)
- (23 баллов)  $d_i \leq 120\,000$ ,  $y_i \leq 120\,000$  (то есть число обезьян в секции не больше 120 000).
- (11 баллов) Нет дополнительных ограничений

## Примеры

стандартный ввод	стандартный вывод
6 5 6 8 7 4 9 1 6 9 0 5	3
4 6 2 3 6 3 3 2 1 3 4 5 1 4 1 1 0 4	1 2 3

## Замечание

В первом примере число обезьян в 6 секции остается равным 9. Таким образом число обезьян в секциях: 5 6 8 7 4 9. Во время дрессировки рассматриваются секции между перегородками 0 и 5, то есть секции с 5, 6, 8, 7, 4 обезьянами. Если убрать перегородки между секциями 1 и 2, и секциями 3 и 4, получим число обезьян равное 11, 15, 4. Это чередующаяся последовательность длины 3. Более длинную получить нельзя.

Рассмотрим дрессировки во втором примере:

- Во время первой дрессировки число обезьян в секции 3 становится равным 2, таким образом число обезьян в секциях: 6 2 2 6. Во время дрессировки рассматриваются обезьяны между перегородками 1 и 3, то есть секции с 2, 2 обезьянами. Из них можно составить чередующуюся последовательность длины 1 (число обезьян: 4).
- Во время второй дрессировки число обезьян в секции 4 становится равным 5, таким образом число обезьян в секциях: 6 2 2 5. Во время дрессировки рассматриваются обезьяны между перегородками 1 и 4, то есть секции с 2, 2, 5 обезьянами. Из них можно составить чередующуюся последовательность длины 2 (число обезьян: 4, 5).
- Во время третьей дрессировки число обезьян в секции 1 становится равным 1, таким образом число обезьян в секциях: 1 2 2 5. Во время дрессировки рассматриваются обезьяны между перегородками 0 и 4, то есть секции с 1, 2, 2, 5 обезьянами. Из них можно составить чередующуюся последовательность длины 3 (число обезьян: 3, 2, 5).