

Задача А. Восстанови максимумы

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

Эта задача с двойным запуском.

Дан массив $v[0 \dots n - 1]$ из попарно различных целых чисел. Требуется сохранить о нём некоторую битовую строку s , чтобы затем по ней отвечать на запросы вида:

для данных a, b ($0 \leq a \leq b < n$) нужно вернуть такой индекс id , что $a \leq id \leq b$ и $v[id] = \max(v[a], v[a + 1], \dots, v[b])$

Иными словами, для каждого запроса нужно вывести индекс максимального элемента на отрезке.

Программа участника запускается два раза.

Протокол взаимодействия

Первый запуск

В первом запуске на вход подаётся сам массив. Нужно построить битовую строку, которая затем будет использована во втором запуске.

Первая строка содержит единственное число 1.

Вторая строка содержит два целых числа $subtask_id, n$

Третья строка содержит n попарно различных целых чисел v_0, v_1, \dots, v_{n-1}

Выведите одну непустую строку s состоящую только из символов 0 и 1.

Если длина этой строки равна L , то должно выполняться $1 \leq L \leq 200000$

Именно эта строка будет передана во втором запуске.

Второй запуск

Во втором запуске сам массив уже **не дан**. Известны только $subtask_id, n$, строка $bits$, выведенная в первом запуске, список запросов.

Для каждого запроса нужно вывести индекс максимального элемента на соответствующем отрезке.

Первая строка содержит единственное число 2.

Вторая строка содержит два целых числа $subtask_id, n$

Третья строка содержит битовую строку s

Четвертая строка содержит число запросов m

Следующие m строк содержат по два целых числа a, b и задают запрос: найти индекс максимального элемента на отрезке $[a, b]$.

Выведите m строк.

В i -й строке должен находиться ответ на i -й запрос: индекс такого элемента id , что $a_i \leq id \leq b_i$ и $v[id] = \max(v[a_i], v[a_i + 1], \dots, v[b_i])$

- элементы массива $v[i]$ попарно различны
- $-10^9 \leq v[i] \leq 10^9$

Система оценки

Баллы за подзадачу начисляются, только если пройдены все тесты этой подзадачи и всех необходимых для неё подзадач.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи
0	0	Тесты из условия	—
1	7	$1 \leq n \leq 500, 0 \leq v[i] < n, 1 \leq m \leq 1000$	—
2	21	$1 \leq n \leq 10000, 1 \leq m \leq 20000$	—
3	72	$1 \leq n \leq 40000, 1 \leq m \leq 80000$	—

В подзадаче 2 оценка каждого теста равна $\min\left(1, \frac{1}{2^{L/n-1-\log_2 n}}\right)$. Оценка за подзадачу равна минимуму оценок по всем тестам этой подзадачи, умноженному на 21.

В подзадаче 3 оценка каждого теста равна $\min\left(1, \frac{1}{2^{L/n-2}}\right)$. Оценка за подзадачу равна минимуму оценок по всем тестам этой подзадачи, умноженному на 72.

Примеры

стандартный ввод	стандартный вывод
1 0 4 40 20 30 10	001101010101
2 0 4 001101010101 10 0 0 0 1 0 2 0 3 1 1 1 2 1 3 2 2 2 3 3 3	0 0 0 0 1 2 2 2 2 3

Задача В. Странные танцы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Елизавета, космическая ведьма, путешествуя на своей метле по галактике, наткнулась на планету, где все жители танцуют странный танец. Эта планета называется Отрадное.

В этом танце участники сначала выстраиваются в ряд, а затем переставляются следующим образом. Пусть танцуют 2^k человек. Тогда человек, стоящий на позиции с двоичной записью $\overline{b_1 b_2 \dots b_k}$, после танца переходит на позицию $\overline{b_k b_{k-1} \dots b_1}$.

Здесь b_1 — старший бит, b_k — младший бит. Позиции нумеруются с 0.

Елизавета также заметила, что каждый житель Отрадного носит одежду одного из 26 цветов. Цвета обозначаются строчными буквами латинского алфавита.

Жители Отрадного называют последовательность цветов *хорошей*, если после описанной перестановки последовательность не изменяется.

Например, при $k = 2$ есть 4 позиции: 0, 1, 2, 3. После танца они идут в порядке 0, 2, 1, 3. Поэтому последовательность *abba* является хорошей, а последовательность *abca* — нет.

Теперь жители Отрадного показывают Елизавете длинный ряд из n танцоров и задают q запросов. Каждый запрос задаётся двумя числами i и k и спрашивает, является ли хорошей подстрока длины 2^k , начинающаяся с позиции i .

Требуется ответить на все запросы.

Формат входных данных

Первая строка содержит строку s длины n ($1 \leq n \leq 5 \cdot 10^5$), состоящую из строчных букв латинского алфавита. i -й символ строки задаёт цвет одежды танцора с номером i .

Каждая из следующих строк содержит два целых числа i, k ($0 \leq i < n, i + 2^k \leq n$) — задаёт запрос: нужно определить, является ли хорошей подстрока строки s , начинающаяся с позиции i и имеющая длину 2^k .

Пусть q — количество запросов, тогда $1 \leq q \leq 5 \cdot 10^5$. Гарантируется, что рассматриваемая подстрока целиком содержится в строке s .

Формат выходных данных

Для каждого запроса выведите 1, если соответствующая подстрока является хорошей, и 0 иначе.

Система оценки

Баллы за подзадачу начисляются, только если пройдены все тесты этой подзадачи и всех необходимых для неё подзадач.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи
0	0	Тесты из условия	—
1	13	$1 \leq n \leq 1000, 1 \leq q \leq 1000$	—
2	37	$1 \leq n \leq 100000, 1 \leq q \leq 100000$	1
3	17	строка s состоит только из символов a и b ; символы выбираются независимо случайным образом, причём для каждого теста вероятности фиксированы	—
4	33	без дополнительных ограничений	1–3

Пример

стандартный ввод	стандартный вывод
аххуххуб	1
0 3	1
1 1	1
0 2	1
3 2	1

Замечание

В запросе (0, 3) рассматривается вся строка `аххуххуб` длины 8; она является хорошей.

В запросе (1, 1) рассматривается подстрока `хх`; она является хорошей.

В запросе (0, 2) рассматривается подстрока `ахху`; она является хорошей.

В запросе (3, 2) рассматривается подстрока `ухху`; она является хорошей.

Задача С. Угадай перестановку

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Тимофей оказался в довольно неприятной ситуации.

Он нашёл старую машину. Внутри машины находятся n скрытых переключателей $s[0], s[1], s[2], \dots, s[n-1]$. Каждый переключатель может быть либо выключен (это обозначается как $s[i] = 0$), либо включён (это обозначается как $s[i] = 1$). Изначально все переключатели выключены.

У машины также есть n кнопок, пронумерованных от 0 до $n-1$, экран и секретная перестановка $p = p[0], p[1], p[2], \dots, p[n-1]$ чисел $0, 1, 2, \dots, n-1$.

На большом плакате на машине написано: «Если ты сможешь угадать перестановку p , то получишь большой приз!»

Тимофей очень хочет получить приз, поэтому просит вас помочь.

Чтобы определить перестановку p , вы можете нажимать кнопки машины. Когда вы нажимаете кнопку с номером i , происходит следующее:

1. Пусть k — количество нажатий любых кнопок до текущего момента.
2. Переключатель с индексом $p[k \bmod n]$ меняет своё состояние, то есть

$$s[p[k \bmod n]] := 1 - s[p[k \bmod n]].$$

3. После этого на экране отображается текущее состояние переключателя с индексом i , то есть значение $s[i]$.

Требуется определить скрытую перестановку p , совершив достаточно малое число нажатий.

Поскольку машина старая, она сломается, если вы нажмёте кнопки слишком много раз. Если число нажатий превысит $50 \cdot n$, вы не получите приз.

Протокол взаимодействия

В начале работы программа получает одно целое число n ($1 \leq n \leq 10^4$) — размер перестановки. После этого ваша программа может делать запросы двух видов.

Чтобы нажать кнопку с номером i , выведите строку вида

? i

где $0 \leq i \leq n-1$.

После этого не забудьте сбросить буфер вывода и считать одно целое число x . Оно равно текущему значению переключателя $s[i]$ после выполнения описанных выше действий. Значение x всегда равно 0 или 1.

Когда ваша программа восстановит перестановку, она должна вывести строку вида

! p[0] p[1] ... p[n-1]

После этого программа должна завершить работу.

После каждого запроса вида ? i и после финального ответа нужно выполнять сброс буфера вывода.

Система оценки

Баллы за подзадачу начисляются, только если пройдены все тесты этой подзадачи и всех необходимых для неё подзадач.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи
0	0	Тесты из условия	—
1	20	$n = 32$	—
2	40	$n = 1000$	—
3	40	$n = 10000$	—

Оценка каждого теста зависит от числа запросов.

Пусть k — количество запросов вида ? i , сделанных вашей программой, а

$$q = \frac{k}{n}.$$

Если перестановка определена неверно или $q > 50$, то за тест начисляется 0 баллов.

Иначе балл за тест равен s , умноженному на максимальный балл подзадачи, где s задаётся следующим образом.

Для подзадачи 1:

$$s = \begin{cases} 1, & q \leq 7, \\ 1 - \frac{1}{10}(q - 7), & 7 < q \leq 12, \\ \frac{1}{2} - \frac{1}{100}(q - 12), & 12 < q \leq 32, \\ \frac{3}{10} - \frac{1}{10} \sqrt[4]{\frac{q-32}{18}}, & 32 < q \leq 50. \end{cases}$$

Для подзадачи 2:

$$s = \begin{cases} 1, & q \leq 13, \\ \frac{1}{2} + \frac{1}{2} \left(\frac{23-q}{10} \right)^2, & 13 < q \leq 23, \\ \frac{1}{5} + \frac{3}{10} \sqrt{\frac{50-q}{27}}, & 23 < q \leq 50. \end{cases}$$

Для подзадачи 3:

$$s = \begin{cases} 1, & q \leq 17, \\ \frac{3}{5} + \frac{25-q}{20}, & 17 < q \leq 25, \\ \frac{1}{5} + \frac{2}{5} \left(\frac{50-q}{25} \right)^2, & 25 < q \leq 50. \end{cases}$$

Пример

стандартный ввод	стандартный вывод
4	? 0
	? 2
1	? 1
1	? 3
1	
1	! 0 2 1 3

Замечание

Не забывайте сбрасывать буфер вывода после каждого запроса. Например, в C++ это можно делать с помощью `cout << endl`; или `cout.flush()`;

Задача D. Cerca Trova

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

Cerca Trova выкатили странноватую, но эффективную новинку: растягивающиеся галстуки. Такой галстук можно удлинять сколько угодно, и именно это Александр собирается продемонстрировать зрителям на сцене.

На сцену выходят k моделей, и изначально длина галстука у каждой из них равна 1. Дальше проходит n номеров шоу, и поскольку необходимо, чтобы шоу было успешным, то некоторых зрителей подкупили. Поэтому Александр точно знает, что в i -м номере зрители выкрикивают число a_i . Каждый раз, когда зрители выкрикивают число a_i , то нужно сделать одно из двух

- **Ответить на число.** Тогда нужно выбрать одну из моделей, у которой текущая длина галстука не больше a_i , и удлинить её галстук до значения a_i (то есть после удлинения значение будет равно a_i). Разрешается выбрать модель, у которой длина уже равна a_i . Если подходящей модели нет, шоу проваливается, и Александр разоряется.
- **Проигнорировать число.** Тогда ничего не происходит.

Но есть важный нюанс: если Александр проигнорирует числа зрителей два раза подряд или больше, публика начнёт свистеть, и шоу тоже провалится, что не допустимо.

Модели стоят денег, так что хочется нанять как можно меньше людей. По известной заранее последовательности чисел a_1, a_2, \dots, a_n найдите минимальное число моделей k , при котором можно провести всё шоу без провала.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 5 \cdot 10^6$) — количество номеров шоу.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 21$) — числа, которые зрители выкрикнут по ходу выступления.

Формат выходных данных

Выведите одно целое число — минимальное количество моделей k , которого достаточно, чтобы шоу не провалилось.

Система оценки

Подгруппа	Баллы	Дополнительные ограничения	Необходимые подгруппы
0	0	Тесты из условия	—
1	10	$n \leq 15$	—
2	6	$n \leq 500, a_i \leq 2$	—
3	12	$n \leq 500, a_i \leq 5$	2
4	18	$n \leq 500, a_i \leq 15$	2, 3
5	26	$n \leq 500\,000, a_i \leq 15$	2, 3, 4
6	10	$n \leq 500\,000$	0–5
7	18	Без дополнительных ограничений	0–6

Примеры

стандартный ввод	стандартный вывод
5 5 3 4 2 1	2
6 2 1 1 2 2 1	1
10 2 4 6 7 4 5 5 3 4 1	3