

## Задача А. Четные и нечетные сочетания

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

**Это задача с двойным запуском.**

*Сочетанием из  $n$  элементов по  $k$*  будем называть  $k$ -элементное подмножество  $n$ -элементного множества  $\{1, 2, \dots, n\}$ . Чтобы записать сочетание, перечислим его элементы в порядке возрастания. Например, сочетания из 3 элементов по 2 выглядят так:  $\{1, 2\}$ ,  $\{1, 3\}$ ,  $\{2, 3\}$ .

Будем называть сочетание *четным*, если количество элементов в нём — четное число, и *нечетным* в противном случае. Зафиксируем  $n > 0$  и рассмотрим два множества:  $A_n$ , множество всех чётных сочетаний из  $n$  элементов, и  $B_n$ , множество всех нечетных сочетаний из  $n$  элементов. Можно доказать, что  $A_n$  и  $B_n$  содержат одинаковое количество сочетаний.

Для каждого  $n = 1, 2, \dots, 50$  задача такова. Постройте любую биекцию (взаимно однозначное соответствие) между множествами  $A_n$  и  $B_n$ . После этого по данному элементу одного из этих множеств выводите соответствующий ему элемент другого множества.

### Формат входных данных

При первом запуске в первой строке записано целое число  $t$  — количество тестовых случаев ( $1 \leq t \leq 1000$ ). Далее следуют их описания.

Каждый тестовый случай задает сочетание и занимает две строки.

В первой из них записаны через пробел два целых числа  $n$  и  $k$  ( $1 \leq n \leq 50$ ,  $0 \leq k \leq n$ ).

Во второй записаны через пробел  $k$  целых чисел  $a_1, a_2, \dots, a_k$  — элементы сочетания ( $1 \leq a_1 < a_2 < \dots < a_k \leq n$ ). Если  $k = 0$ , то вторая строка пуста.

При втором запуске формат ввода точно такой же, как при первом запуске. Но в каждом тестовом случае дано не исходное сочетание, а то, которое было выведено при первом запуске.

### Формат выходных данных

При первом запуске в ответ на каждый тестовый случай выведите сочетание из другого множества, соответствующее заданному. Формат сочетания в выводе — точно такой же, как во вводе. У выведенного сочетания  $n$  должно совпадать с заданным, а  $k$  должно иметь другую четность. Других ограничений на выбор соответствия нет.

При втором запуске в ответ на каждый тестовый случай, как и при первом запуске, выведите сочетание из другого множества, соответствующее заданному. Формат сочетания в выводе — точно такой же, как во вводе. Поскольку соответствие должно быть взаимно однозначным, выведенное при втором запуске сочетание должно совпадать с тем, которое было дано при первом запуске. Это и будет проверять программа жюри.

### Протокол взаимодействия

Для проверки того, что вы действительно построили биекцию, в этой задаче ваше решение будет запущено на каждом тесте два раза. В конце каждой строки входных данных следует символ перевода строки.

### Пример

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, в зависимости от вашего вывода на первом запуске, ввод на втором запуске может быть другим.

стандартный ввод	стандартный вывод
6 3 0  2 1 1 3 3 1 2 3 3 1 1 3 1 2 3 1 3	3 3 1 2 3 2 2 1 2 3 0  3 2 2 3 3 2 1 3 3 2 1 2
6 3 3 1 2 3 2 2 1 2 3 0  3 2 2 3 3 2 1 3 3 2 1 2	3 0  2 1 1 3 3 1 2 3 3 1 1 3 1 2 3 1 3

## Задача В. Тождество

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Строкой Фибоначчи длины  $n$  называется последовательность из  $n$  нулей и единиц, в которой нет двух единиц подряд. Число различных строк Фибоначчи длины  $n$  будем записывать как  $F_n$ . и называть числом Фибоначчи. Известно, что  $F_n = F_{n-1} + F_{n-2}$ . Обратите внимание: последовательность начинается с  $F_0 = 1$  и  $F_1 = 2$ , чем отличается от классической нумерации чисел Фибоначчи! Например, при  $n = 3$  существует пять различных строк Фибоначчи: это «000», «001», «010», «100» и «101».

Сочетанием из  $m$  элементов по  $k$  называется последовательность целых чисел  $a_1, a_2, \dots, a_k$ , в которой все числа лежат в пределах от 1 до  $m$  и записаны в порядке строгого возрастания. Число различных сочетаний из  $m$  элементов по  $k$  записывается как  $C_m^k$ . Известно, что  $C_m^k = C_{m-1}^{k-1} + C_{m-1}^k$ , при этом  $C_0^0 = 1$ , а кроме того,  $C_m^k = 0$  при  $k > m$ . Например, при  $m = 4$  и  $k = 2$  существует шесть различных сочетаний: это (1, 2), (1, 3), (1, 4), (2, 3), (2, 4) и (3, 4).

Недавно Ада прочитала в книге следующее тождество, связывающее эти объекты:

$$F_n = C_{n+1}^0 + C_n^1 + C_{n-1}^2 + C_{n-2}^3 + \dots$$

Суммирование ведётся, пока верхний индекс числа сочетаний не превосходит нижнего: далее все значения равны нулю. Например, при  $n = 3$  тождество принимает вид

$$F_3 = C_4^0 + C_3^1 + C_2^2.$$

Ада хочет показать тождество своему другу Чарли. Она придумала, как доказать тождество по индукции. Но Чарли не любит формальные рассуждения: ему больше по вкусу конкретные примеры и наглядные доказательства. Чарли сможет по достоинству оценить тождество, если построить биекцию: взаимно однозначное соответствие одних объектов другим. Зафиксируем число  $n$ . Каждой строке Фибоначчи длины  $n$  следует сопоставить одно из сочетаний, перечисленных в правой части равенства: либо сочетание из  $n + 1$  элемента по 0, либо из  $n$  элементов по 1, либо из  $n - 1$  элемента по 2, и так далее.

Помогите Аде построить биекцию для Чарли. Напишите программу, которая любой строке Фибоначчи длины  $n$  сопоставит либо сочетание из  $n + 1$  элемента по 0, либо из  $n$  элементов по 1, либо из  $n - 1$  элемента по 2, и так далее. Затем, получив параметры сочетания и само сочетание, программа должна восстановить исходную строку Фибоначчи.

### Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. При вводе и выводе числа в строках отделяются друг от друга пробелами. В конце каждой строки входных данных следует символ перевода строки.

При первом запуске решение по строке Фибоначчи строит сочетание. В первой строке записано слово «first». Вторая строка содержит целое число  $n$  — длину заданной строки Фибоначчи ( $1 \leq n \leq 300\,000$ ). В третьей строке записана сама строка Фибоначчи —  $n$  двоичных цифр, среди которых нет двух единиц подряд.

В первой строке выведите два целых числа  $m$  и  $k$ : параметры сочетания (должно быть выполнено  $m + k = n + 1$ ). Во второй строке выведите само сочетание:  $k$  различных целых чисел от 1 до  $m$ , перечисленных в порядке возрастания. Если  $k = 0$ , вторую строку (пустую в этом случае) можно как вывести, так и пропустить. Как именно сочетание зависит от заданной строки Фибоначчи — решать вам.

При втором запуске решение по параметрам сочетания и самому сочетанию восстанавливает строку Фибоначчи. В первой строке записано слово «second». Вторая строка содержит целые числа  $m$  и  $k$  — параметры сочетания. В третьей строке задано само сочетание —  $k$  различных целых чисел от 1 до  $m$ , перечисленных через пробел в порядке возрастания. Все эти числа — ровно те, которые решение вывело при первом запуске. Если  $k = 0$ , третья строка входных данных пуста.

В первой строке выведите  $n$  — длину исходной строки Фибоначчи (должно быть выполнено  $m + k = n + 1$ ). Во второй строке выведите строку Фибоначчи длины  $n$ , совпадающую с исходной.

После вывода не забудьте выполнить операцию `flush`! В противном случае вы можете получить произвольный вердикт.

## Система оценивания

Баллы за подзадачу начисляются только в том случае, если все тесты этой подзадачи и всех предыдущих подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Информация о проверке
1	40	$n \leq 10$	До первой ошибки
2	35	$n \leq 40$	До первой ошибки
3	15	$n \leq 1000$	До первой ошибки
4	10	Дополнительных ограничений нет	До первой ошибки

## Примеры

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, в зависимости от вашего вывода на первом запуске, ввод на втором запуске может быть другим.

### Пример 1

Первый запуск.

стандартный ввод	стандартный вывод
first	2 2
3	1 2
101	

Второй запуск.

стандартный ввод	стандартный вывод
second	3
2 2	101
1 2	

### Пример 2

Первый запуск.

стандартный ввод	стандартный вывод
first	4 0
3	
000	

Второй запуск.

стандартный ввод	стандартный вывод
second	3
4 0	000

## Задача С. 1.5G — модем

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Алиса должна передать Бобу целое неотрицательное число  $X$ , меньшее  $10^{10}$ . Единственным способом связи у Алисы является старый 1.5G модем, который при передаче произвольным образом переставляет цифры в числе.

Алиса может отправить по модему число из не более, чем 2021 цифр (ведущие нули допустимы).

Боб получает от Алисы отправленное число с переставленными произвольным образом цифрами. Задача Боба — восстановить по полученному сообщению число  $X$ .

### Формат входных данных

В первой строке входных данных задаётся одна строка — «Alice», если данные предназначены для Алисы, и «Bob», если данные предназначены для Боба.

В случае, когда данные предназначены для Алисы, вторая строка содержит целое неотрицательное число  $X$ , меньшее  $10^{10}$ , заданное без ведущих нулей.

В случае, когда данные предназначены для Боба, вторая строка содержит строку  $s$  из цифр — отправленное Алисой число с переставленными цифрами. Строка не может быть пустой или состоять из более чем 2021 цифры.

### Формат выходных данных

Если входные данные предназначены для Алисы, выведите непустую строку, состоящую из не более, чем 2021 цифр.

Если входные данные предназначены для Боба, выведите одно число — предполагаемое значение  $X$ .

### Протокол взаимодействия

Ваше решение будет запущено два раза — в первый раз на данных, предназначенных для Алисы, во второй раз — на сгенерированных первым запуском данных, предназначенных для Боба. Передача каких-либо ещё данных между запусками запрещена.

### Система оценивания

Каждый тест оценивается независимо в 2 балла.

Если тест не решен (то есть вы вывели пустую строку, или строку, содержащую не только цифры, или слишком длинную строку, действуя за Алису, или не угадали число, действуя за Боба) вы получаете 0 баллов.

Если тест решен корректно и длина переданной Алисой строки  $L$  не превосходит 50, вы получаете полный балл за тест, то есть 2 балла.

Иначе вы получаете  $\frac{100}{L}$  баллов за тест.

### Пример

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, в зависимости от вашего вывода на первом запуске, ввод на втором запуске может быть другим.

#### Пример 1

Первый запуск.

стандартный ввод	стандартный вывод
Alice 2021	66889

Второй запуск.

стандартный ввод	стандартный вывод
Bob 68986	2021

## Задача D. Исправление одной ошибки

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Это задача с двойным запуском.

Ваше решение будет запущено два раза.

При первом запуске ему на вход будет передана строка  $x$  из нулей и единиц длиной не больше  $n$ . Программа должна вывести строку  $y$  из нулей и единиц, длина которой не более чем  $m$ . Число  $m$  не известно вашей программе и зависит от подзадачи, соответствующее значение указано в таблице системы оценивания. Если ваша программа выведет строку длиннее, чем  $m$ , она получит вердикт «Wrong answer».

Между запусками решения программа жюри внесет в строку  $y$  не более одной модификации, заменив ноль на единицу или единицу на ноль, получив, таким образом, строку  $z$ .

При втором запуске программе на вход будет подана строка  $z$ . Она должна восстановить исходную строку  $x$  и выдать её на выход.

### Формат входных данных

При первом запуске на первой строке ввода находится число 1. На второй строке ввода находится строка  $x$  из нулей и единиц длины  $n$  ( $10 \leq n \leq 100\,000$ ).

При втором запуске на первой строке ввода находится число 2. На второй строке ввода находится строка  $z$  из нулей и единиц длины не больше  $m$  ( $10 \leq m \leq 300\,000$ ). Гарантируется, что эта строка равна строке  $y$ , выведенной программой при первом запуске, или получена из неё изменением ровно одного символа на противоположный.

### Формат выходных данных

При первом запуске необходимо вывести строку  $y$ , которая позволит восстановить  $x$  после внесения в нее изменения. Длина строки  $y$  не должна превышать  $m$ .

При втором запуске по заданной строке  $z$  необходимо восстановить исходную строку  $x$  и вывести её.

### Система оценивания

- 40 баллов.  $n = 10$ ,  $m = 30$
- 15 баллов.  $n = 100\,000$ ,  $m = 300\,000$
- 15 баллов.  $n = 100\,000$ ,  $m = 200\,000$
- 15 баллов.  $n = 100\,000$ ,  $m = 101\,000$
- 15 баллов.  $n = 100\,000$ ,  $m = 100\,017$

## Примеры

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

### Пример 1

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	00000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 00000000000000000000000000000000	0000000000

### Пример 2

Первый запуск.

стандартный ввод	стандартный вывод
1 0000000000	00000000000000000000000000000000

Второй запуск.

стандартный ввод	стандартный вывод
2 00000000000010000000000000000000	0000000000

## Задача Е. Прерванная сортировка

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

**Это задача с двойным запуском.**

Петя — это робот, основной задачей которого является сортировка массивов. В его памяти хранится массив длины  $n$  (от 1 до 100), элементы которого являются попарно различными целыми числами. Элементы массива пронумерованы числами от 1 до  $n$  слева направо.

Нина является оператором данного робота. Нина хочет отсортировать массив, хранящийся в памяти робота. Массив называется отсортированным, если для любых двух номеров элементов  $i$  и  $j$  ( $i < j$ )  $i$ -й элемент массива меньше, чем  $j$ -й элемент массива.

Единственная команда, доступная для выполнения роботом, это сравнение двух элементов на позициях  $i$  и  $j$ . Если элемент на левой позиции (ей может оказаться как  $i$ , так и  $j$ ) больше, чем элемент на правой позиции, Петя меняет данные элементы местами и выводит число 1 на экран. В противном случае Петя ничего не делает и выводит число 0 на экран.

К сожалению, аккумулятор Пети поврежден, поэтому иногда робот выключается. Данное событие выглядит следующим образом: вместо исполнения очередной команды Петя выводит число  $-1$  на экран и игнорирует все последующие команды.

Для того, чтобы включить Петю, Нина разбирает его, заменяет аккумулятор и собирает снова. При этом массив, хранящийся в памяти, не меняется. К сожалению, во время ремонта Нина забывает, какие команды робот выполнял ранее. После ремонта Петя работает до тех пор, пока его аккумулятор не разрядится. После этого он снова выключается.

Аккумулятора Пети достаточно, чтобы выполнить 1 500 команд. Во время данного процесса Петя выключается ровно дважды: после выполнения  $x$ -й команды и после выполнения 1 500-й команды ( $0 < x < 1 500$ , значение  $x$  не известно Нине). Зная это, помогите Нине выдать роботу необходимые команды таким образом, чтобы после его второго выключения массив оказался отсортированным.

### Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. Ваше решение играет за Нину, а программа жюри — за Петю.

Ниже описан протокол взаимодействия с программой жюри. Протокол одинаков для обоих запусков.

Первая строка сходных данных содержит целое число  $n$  ( $1 \leq n \leq 100$ ) — размер массива. После этого ваше решение должно выводить команды, а программа жюри будет выполнять их и выдавать результаты выполнения команд.

Для того, чтобы выполнить команду «сравнить элементы на позициях  $i$  и  $j$ », выведите строку вида « $i j$ », где  $1 \leq i, j \leq n$ . В качестве результата выполнения команды вы получите одно целое число:

- Число 1 означает, что Петя поменял элементы на позициях  $i$  и  $j$  местами, потому что левое число было больше, чем правое.
- Число 0 означает, что Петя не менял элементы местами, потому что левое число было не больше, чем правое (то есть либо левое число было меньше, чем правое, либо  $i = j$ ).
- Число  $-1$  означает, что Петя выключился вместо выполнения команды.

В последнем случае ваше решение должно немедленно завершить свою работу. В остальных случаях вы можете запросить выполнение очередной команды.

Если вы хотите прекратить взаимодействие с Петей до того, как он выключится, вместо команды выведите строку « $-1 -1$ ». После этого ваше решение должно немедленно завершить свою работу.

В каждом тесте массив зафиксирован заранее перед первым запуском. Массив перед вторым запуском находится в состоянии, в котором он был в конце первого запуска. Также в каждом тесте число команд  $x$ , после которых Петя выключится в первый раз ( $0 < x < 1 500$ ) зафиксировано



заранее. Во время второго запуска робот выключится после  $1500 - x$  команд, даже в том случае, если во время первого запуска взаимодействие с Петей было завершено до его выключения.

### Пример

стандартный ввод	стандартный вывод
5	1 5
1	1 2
1	2 3
0	3 4
1	4 5
1	2 1
0	3 2
1	4 3
0	1 2
-1	
5	1 2
1	2 3
0	1 2
0	-1 -1

## Задача F. Сообщение из шума

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

**Это задача с двойным запуском.**

Алиса хочет передать Еве по *числовому проводу* сообщение — одно английское слово.

К сожалению, сейчас числовой провод передаёт только шум: случайные целые числа от 0 до  $10^9 - 1$  включительно. Алисе известна последовательность из следующих 10 000 чисел, которые будут переданы.

К счастью, у Алисы есть суперспособность: стереть из этой последовательности любое количество элементов на любых позициях. Порядок оставшихся чисел не поменяется.

К сожалению, после этого примерно половина чисел потеряется при передаче: каждое передаваемое число с вероятностью  $\frac{1}{2}$  исчезнет. Порядок оставшихся чисел опять не поменяется.

Как должны действовать Алиса и Ева, чтобы передать заданное слово?

### Формат входных данных

В этой задаче ваше решение будет запущено на каждом тесте два раза. При вводе и выводе числа в одной строке отделяются друг от друга пробелами. В конце каждой строки входных данных следует символ перевода строки.

### Формат выходных данных

При первом запуске решение действует за Алису. В первой строке записано имя «Alisa». Вторая строка содержит одно слово из английского словаря, имеющее длину от 2 до 15 букв и состоящее из маленьких букв. В третьей строке записано целое число  $n$  — размер последовательности (в этой задаче  $n$  всегда равно 10 000). Четвертая строка содержит  $n$  чисел от 0 до  $10^9 - 1$  включительно — исходную последовательность. Числа выбраны заранее генератором псевдослучайных чисел, все целые числа из диапазона равновероятны.

При втором запуске решение действует за Еву. В первой строке записано имя «Eva». Во второй строке записано целое число  $k$  — количество чисел, оставшихся в последовательности. Третья строка содержит  $k$  чисел от 0 до  $10^9 - 1$  включительно — то, что осталось от последовательности: каждое число, которое Алиса решила оставить в первом запуске, с вероятностью  $\frac{1}{2}$  оказалось здесь и с вероятностью  $\frac{1}{2}$  исчезло. Как именно исчезают числа из последовательности — зафиксировано заранее в каждом тесте, то есть, если решения ведут себя одинаково при первом запуске, то они получат одинаковые последовательности при втором запуске.

### Протокол взаимодействия

При первом запуске вывести следует те числа, которые Алиса решила **оставить** в последовательности. В первой строке выведите целое число  $m$  — количество оставшихся чисел. Во второй строке выведите сами эти числа в том же порядке, в котором они следуют в исходной последовательности.

При втором запуске выведите одно английское слово — то, которое Алиса должна была передать Еве.

### Пример

Обратите внимание, что в примерах приведены конкретные варианты вывода и ввода при втором запуске, в зависимости от вашего вывода на первом запуске, ввод на втором запуске может быть другим.

Для краткости последовательности в примере показаны не полностью. Полную версию примера можно найти в `samples.zip`.

стандартный ввод	стандартный вывод
Alisa spark 10000 833080662 16249270 933346436 811379468 <...> 13286897 459644281	3900 933346436 811379468 877083772 408973036 <...> 583178591 13286897
Eva 1955 811379468 408973036 585189166 111199534 <...> 226510051 829146141	spark