

Задача А. К-близнецы

Имя входного файла: `twins.in`
Имя выходного файла: `twins.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В теории чисел простыми числами-близнецами называют пару таких простых чисел (p, q) , что $q - p = 2$. Например, пары $(3, 5)$ и $(11, 13)$ являются парами простых чисел-близнецов. Назовем *обобщенными числами-близнецами* пару простых чисел (p, q) , где $q - p = k$, k — некоторое натурально-е число. Например, для $k = 4$ пара $(3, 7)$ является парой обобщенных чисел-близнецов. Существует предположение, что пар простых чисел-близнецов бесконечно много, однако это не доказано. Безусловно, выяснить по заданному k , сколько пар обобщенных близнецов содержит множество всех натуральных чисел, не менее сложная задача, чем аналогичная о простых близнецах. Ваша же задача несколько проще — выяснить по заданному k , сколько пар обобщенных близнецов содержит множество натуральных чисел от 1 до n .

Формат входных данных

В первой строке входного файла через пробел заданы два натуральных числа n и k ($1 \leq n, k \leq 10^4$).

Формат выходных данных

В выходной файл выведите число пар простых чисел (p, q) , таких, что $1 \leq p < q \leq n$ и $q - p = k$.

Примеры

<code>twins.in</code>	<code>twins.out</code>
17 2	3
10000 1	1

Задача В. Коварное ДП

Имя входного файла: `seq.in`
Имя выходного файла: `seq.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дмитрий Витальевич любит давать сложные задачи студентам. Один раз он перестарался, поэтому результаты первокурсников резко стали ухудшаться. Дмитрий Витальевич считает, что студент успешен, если его оценки строго возрастают.

Загробленностью курса для конкретного студента преподаватель считает длину самой большой подпоследовательности в его оценках, которая *не является возрастающей*.

По заданной последовательности оценок студента найдите *загробленность курса* для него.

Подпоследовательность с индексами $i_1 < i_2 < \dots < i_k$ называется *возрастающей*, если $a[i_1] < a[i_2] < \dots < a[i_k]$.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 100$).

Во второй строке записаны n целых чисел, оценок студента, каждое из которых лежит в диапазоне от 0 до 100.

Формат выходных данных

В первой строке выведите единственное целое число — *загробленность курса*. Во второй строке выведите *индексы* элементов найденной последовательности.

Пример

<code>seq.in</code>	<code>seq.out</code>
3	3
1 3 2	1 2 3

Задача С. Строка Леонардо Пизанского

Имя входного файла: `fib1.in`
Имя выходного файла: `fib1.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Леонардо Пизанский, также известный под прозвищем Фибоначчи, любил выписывать разные строки, а потом смотреть на них (а вот вы чем полезным занимаетесь на Новый Год?). В один момент он стал записывать *строки Фибоначчи*.

Легко догадаться, что строки эти заданы в виде рекуррентного соотношения. Пусть F_i — i -я строка Фибоначчи.

$$F_0 = a, F_1 = b, F_i = F_{i-2}F_{i-1}, i > 1$$

В один момент Леонардо надоело писать строки, поэтому он просит вас просто определить, какой символ стоял в строке F_k на позиции p по заданным k и p .

Формат входных данных

Входной файл содержит несколько наборов входных данных. Первая строка входного файла содержит целое число T наборов входных данных ($1 \leq T \leq 100$). Каждая из последующих T строк описывает один набор входных данных и содержит по два целых числа: n и k ($0 \leq n \leq 45, 1 \leq k \leq |F_n|$), как $|F_n|$ обозначена длина строки F_n , позиции символов в строке нумеруются с единицы).

Формат выходных данных

Выведите в выходной файл T строк, каждая из которых должна содержать ровно один символ — ответ для соответствующего набора входных данных.

Пример

<code>fib1.in</code>	<code>fib1.out</code>
4	a
0 1	b
1 1	a
3 2	a
7 7	

Задача D. Враг моего врага

Имя входного файла: `enemy.in`
Имя выходного файла: `enemy.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Винни-Пуху очень понравилась эта социальная сеть. Он целыми днями сидел и записывал, кто же становился чьим врагом, так как хотел знать все, что происходит в их лесу. Он считал, что никто не пойдет в гости к своему врагу. Также, по его мнению, враг врага является другом, а любая уважающая своих друзей персона должна пойти в гости к своему другу. Винни-Пуху очень интересно узнать — сколько же у пользователя под номером v друзей. Пользователь u является другом пользователя v по версии Винни-Пуха, если выполняются некоторые условия:

1. u является врагом некоторого врага v
2. u не является врагом v

Заметим также, что никакой пользователь сам не является своим другом.

Формат входных данных

В первой строке входного файла задано числа n и m ($1 \leq n, m \leq 2000$) — количество пользователей, зарегистрированных в социальной сети и количество запросов соответственно.

В следующих m строках заданы запросы двух видов:

1. `+ v u` — пользователь v начал считать пользователя u своим врагом
2. `? v` — узнать количество друзей пользователя v по версии Винни-Пуха

Гарантируется, что входные данные корректны — пользователь не начнет считать себя своим врагом и никакой пользователь не станет врагом другого более одного раза.

Формат выходных данных

Для каждого запроса `? v` выведите одно целое число — ответ на него в отдельной строке.

Пример

<code>enemy.in</code>	<code>enemy.out</code>
5 5	1
+ 1 2	
+ 2 4	
+ 2 5	
+ 1 5	
? 1	