

## Задача А. Строки Фибоначчи

Имя входного файла: `fibstring.in`  
Имя выходного файла: `fibstring.out`  
Ограничение по времени: 2 секунд  
Ограничение по памяти: 256 мегабайт

Алла очень любит палиндромы. Все потому, что её имя является палиндромом. Напомним, что строку называют палиндромом тогда, когда она одинаково читается как слева направо, так и справа налево.

Однажды в школе учитель рассказал Алле про так называемые строки Фибоначчи.

Строки Фибоначчи определяются следующим образом:

- $f_0 = a$
- $f_1 = b$
- $f_n = f_{n-1}f_{n-2}$  для каждого  $n \geq 2$  — конкатенация двух предыдущих строк Фибоначчи

Таким образом, первые шесть строк Фибоначчи: «a», «b», «ba», «bab», «babba».

Аллу сразу заинтересовал вопрос — какой максимально длинный палиндром встречается в  $k$ -й строке Фибоначчи. Помогите Алле решить эту задачу.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $k$  ( $0 \leq k \leq 80$ ) — номер строки Фибоначчи.

### Формат выходных данных

В выходной файл выведите длину самого большого палиндрома, содержащегося в  $k$ -й строке Фибоначчи.

### Примеры

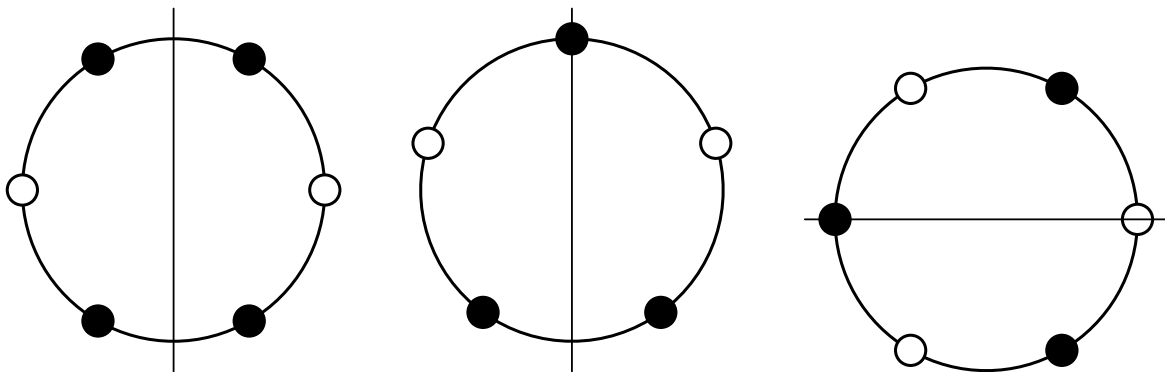
<code>fibstring.in</code>	<code>fibstring.out</code>
2	1
4	4

## Задача В. Симметричная поляна

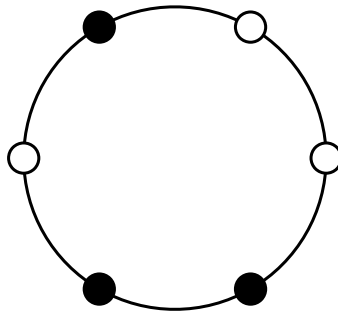
Имя входного файла: `glade.in`  
Имя выходного файла: `glade.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти:

Лесов много, но не каждый из них — волшебный. Для того, чтобы лес был волшебным, должно соблюдаться много условий. Все они еще толком не изучены, но про одно известно достоверно: лес не может быть волшебным, если в нем нет круглой симметричной поляны.

Рассмотрим, для простоты, лес, в котором растут березы и сосны. В этом лесу есть только одна круглая поляна. Вокруг нее растут  $n$  деревьев. Каждое дерево является или сосной, или березой. Ось поляны — прямая, проходящая через центр поляны и разбивающая ее на две половины. Если она проходит через какое-то дерево, то оно попадает в обе половины. Если при этом одна из половин симметрична другой, то это ось симметрии и поляна симметрична. Таким образом, следующие поляны симметричны (белыми кругами обозначены березы, черными — сосны):



А эта поляна не симметрична, какую бы ось мы не выбрали:



### Формат входных данных

В первой строке входного файла содержится одно целое число  $T$  ( $1 \leq T \leq 10$ ) — количество тестов в файле. Далее следуют  $T$  блоков с описаниями самих тестов.

Каждое описание теста состоит из двух строк. Первая строка содержит одно целое число  $n$  ( $2 \leq n \leq 10^5$ ) — количество деревьев, окружающих поляну. В следующей строке содержатся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 1$ ) — описание деревьев, окружающих поляну, в порядке обхода по часовой стрелке. 0 соответствует березе, 1 — сосне.

### Формат выходных данных

Выведите в выходной файл  $T$  строк, содержащих ответы на тесты. В строке с номером  $i$  выведите Yes, если у соответствующей поляны есть ось симметрии, и No — если поляна не симметрична.

## Примеры

glade.in	glade.out
4	Yes
6	Yes
0 1 1 0 1 1	Yes
5	No
0 1 0 1 1	
6	
1 0 1 0 1 0	
6	
0 0 1 0 1 1	

## Примечание

Решения, работающие для  $n \leq 1000$  будут оцениваться в 40 баллов.

## Задача С. Белочка

Имя входного файла: `squirell.in`  
Имя выходного файла: `squirell.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти:

В волшебном лесу растут волшебные деревья. Чтобы деревья могли разговаривать друг с другом, у каждого дерева есть своё имя. Они не хотят беспокоить друг друга зря, поэтому, все их имена различны.

В волшебных деревьях в волшебных дуплах живут волшебные белочки. У каждого дупла есть свой номер, и, чем выше дупло, тем больше его номер. Для того, чтобы белочки не заблудились, мудрая сова дала каждой белочке волшебную бумажку с её адресом, на котором были написаны имя дерева и номер дупла в нём. Также, мудрая сова записывает адреса всех белочек в свой волшебный блокнот.

Иногда в лесу появляются новые белочки и хотят поселиться в своё дупло. Каждая из них уже выбрала дерево, в котором хочет жить, но некоторые дупла в нём были уже заняты. Тогда белочка обращается за помощью к мудрой сове, чтобы она подсказала ей свободное дупло в этом дереве. Так как все белочки ленивые, они хотят, чтобы дупло было как можно ниже. Так как дерево волшебное, у него есть сколько угодно дупел, и все они находятся на разной высоте. Помогите сове выбрать белочке дупло.

### Формат входных данных

В первой строке задано число  $n$  — количество записей у совы в блокноте. Во второй строке задано имя дерева, в которое хочет заселиться белочка. Далее, в  $n$  строках заданы записи совы. Одна запись совы состоит из конкатенации имени дерева и номера дупла, в котором живёт белочка. Имя дерева состоит из натурального числа маленьких латинских букв. Номер дупла — натуральное число до  $10^9$ . Размер входного файла

### Формат выходных данных

Выведите единственное число — номер самого низкого дупла, в которое может заселиться белочка.

### Примеры

squirell.in	squirell.out
5 stepan stepan1 vasiliy4 stepan4 stepan3 vladimir1000000	2

### Примечание

Решения, работающие для  $n \leq 1000$  будут оцениваться в 60 баллов.

## Задача D. Легкое слово

Имя входного файла: `word.in`  
Имя выходного файла: `word.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти:

Доминика Петрова прилетела в США из Украины с целью получить гражданство и начать лучшей жизнью. В Америке она вышла замуж за Грегори Хауса и стала Доминикой Хаус. Поначалу ей было очень сложно освоиться в штатах. Еще бы, ведь фиктивный муж не очень стремился ей помогать. Поэтому Доминике пришлось самой изучать сложный английский язык.

По мере того, как Доминика узнавала все новые и новые слова, она поняла, что некоторые слова ей учить проще. Поэтому она ввела понятие сложности слова. Сложность слова — число, равное сумме двух параметров: отклонение слова от первой буквы и от последней. Отклонение слова от буквы  $s$  считается так: для каждой буквы считается модуль разности его позиции в алфавите с позицией буквы  $s$ , после чего берется максимум из всех этих величин.

Также Доминика заметила, что после нескольких циклических сдвигов слово переходит само в себя. Теперь ей стало интересно, сколько циклических сдвигов с минимальной сложностью ей встретится до того, как сдвиг строки первый раз совпадет с исходной строкой.

### Формат входных данных

Во входном файле дано одно слово, состоящее из строчных букв английского алфавита. Длина слова не превосходит  $10^6$ .

### Формат выходных данных

В выходной файл выведите два числа: минимальную сложность циклического сдвига, который встретится Доминике до того, как строка перейдет в себя, и число таких сдвигов.

### Примеры

<code>word.in</code>	<code>word.out</code>
abcbabc	3 2