

## Задача А. Обобщенные бурундуки Фибоначчи

Имя входного файла: `fibonacci.in`  
 Имя выходного файла: `fibonacci.out`  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

Как известно, бурундуки живут  $a$  лет. Бурундуки — существа любвеобильные, поэтому каждый год среди них рождаются новые. А именно, если в год  $y$  в каком-либо месте всего было  $b$  бурундуков, то в год  $y + 1$  родится еще  $b$ .

Недавно Вася ограбил зоомагазин. Так как Вася достаточно хорошо разбирается в бурундуках, внимательно изучив конкретное животное, он может определить его возраст. Именно этим грабитель и занялся в первый же день знакомства со своей добычей. К его удивлению, среди зверей оказалось ровно  $n_1$  бурундуков, которым всего 1 год, ровно  $n_2$  бурундуков возрастом в 2 года, и так далее.

Теперь Вася заинтересовался, сколько же бурундуков у него будет через  $k$  лет. А так как у него со школьных лет большая нелюбовь к большим числам, ему намного больше хочется увидеть это число по модулю  $M$ .

### Формат входного файла

В первой строке входного файла находится число  $a$  ( $1 \leq a \leq 5$ ) — срок жизни бурундуков. Далее, в следующих  $a$  строках находятся числа  $n_1, n_2, \dots, n_a$  ( $0 \leq n_i \leq 100$ ) — количества бурундуков возрастом в 1, 2, ...,  $a$  лет соответственно.

В последней строке входного файла находится два числа:  $k$  и  $M$  ( $0 \leq k \leq 10^9, 1 < M \leq 10^9$ ) — интересующий промежуток времени, и модуль, по которому требуется вывести количество бурундуков.

### Формат выходного файла

Выведите в выходной файл единственное число — количество бурундуков через  $k$  лет по модулю  $M$ .

### Пример

<code>fibonacci.in</code>	<code>fibonacci.out</code>
2	13
0 1	
5 10000	

## Задача В. Folding

Имя входного файла: `folding.in`  
 Имя выходного файла: `folding.out`  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

Билл пытается компактно представить последовательность заглавных латинских букв от  $A$  до  $Z$ , с учетом повторяющихся последовательностей в ней. Например, возможный способ представить последовательность `AAAAAAAAAABABABCCD` — это  $10(A)2(BA)2(C)D$ . Билл ввел формальное понятие упакованной последовательности так:

- Последовательность, содержащая один символ из диапазона  $A \dots Z$ , считается упакованной последовательностью. Ее распаковка возвращает тот же символ.
- Если  $S$  и  $Q$  — упакованные последовательности, то  $SQ$  — также упакованная последовательность. Причем, если результатом распаковки  $S$  является  $S'$ , а  $Q = Q'$ , то  $SQ$  распаковывается в  $S'Q'$ .
- Если  $S$  — упакованная последовательность, то  $X(S)$  — также упакованная последовательность, где  $X$  — десятичное целое число, большее 1. Если  $S$  распаковывается в  $S'$ , то  $X(S)$  распаковывается в  $S'$ , повторенную  $X$  раз.

Согласно этому определению легко распаковать любую запакованную последовательность. На Билл более заинтересован в обратной операции. Он хочет запаковать данную последовательность так, чтобы результирующая запакованная последовательность содержала как можно меньше символов (включая цифры и скобки).

### Формат входного файла

Входной файл содержит одну строку, состоящую не менее, чем из одного и не более чем из 100 символов в диапазоне  $'A' \dots 'Z'$ .

### Формат выходного файла

Запишите в выходной файл одно число — длину кратчайшего из вариантов упаковки исходной последовательности.

### Пример

<code>folding.in</code>	<code>folding.out</code>
<code>AAAAAAAAAABABABCCD</code>	<code>9(A)3(AB)CCD</code>
<code>AZOGZOGAZOGZOG</code>	<code>2(A2(ZOG))</code>

## Задача С. Передай другу

Имя входного файла: `friends.in`  
 Имя выходного файла: `friends.out`  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

ЛКШ 2011 будет проходить в Рио де Жанейро, где все ходят с белыми бейджиками. Вы не знали этого? А ведь именно такого рода слухи распространяет среди своих друзей лкшюнок Остап.

Сообщая слух другу, Остап также просит его передать этот слух всем его друзьям. Однако те уже не будут передавать этот слух далее. Например, если Аня и Ваня — друзья, Ваня и Саня — друзья, а Аня и Саня — не друзья, и Остап сообщает какой-то слух Ане, то новость дойдет до Вани, но не до Сани.

Каждому человеку Остап может сообщить не более одного слуха (иначе его уличат в выдумке). Какое наибольшее число слухов Остап может распространить, так чтобы каждый из них дошел до каждого его друга?

### Формат входного файла

Во входном файле находится квадратная таблица  $n \times n$ , где  $n \leq 15$  — число друзей Остапа. В  $i$ -й строке в  $j$ -

м столбце содержится символ Y, если  $i$ -й и  $j$ -й друзья Остапа дружат между собой, и N — иначе.

### Формат выходного файла

Выведите наибольшее количество слухов, которые может распространить Остап среди всех своих друзей.

#### Пример

friends.in	friends.out
NYYN YNYY YYNY NYYN	3
NYYN YNYN YYNN NNNN	1
NYNNNY YNYNNN NYNYNN NNYNYN NNNYNY YNNNYN	3

### Задача D. Экспериментатор

Имя входного файла: `tester.in`  
 Имя выходного файла: `tester.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

На старости лет один профессор загорелся идеей исследования на прочность транзисторов «КД521(2)». К сожалению, ему не удалось привлечь на помощь никого из коллег, поэтому проводить измерения придется самостоятельно. Но это не пугает профессора.

В шкафу профессор обнаружил  $m$  транзисторов данной модели, оставшихся со старых времен, и решил использовать их для экспериментов.

После некоторых размышлений был выбран следующий способ проведения измерений: профессор собирается, перемещаясь по пожарной лестнице, сбрасывать транзисторы с различных этажей. Таким образом он планирует определить, при падении с какого минимального этажа транзистор разбивается. При этом профессор уверен, что транзистор не может выдержать падение с последнего этажа, однако падение с высоты человеческого роста (то есть когда профессор находится на первом этаже) не причиняет транзистору вреда.

Известно, что все транзисторы абсолютно одинаковые, и если транзистор разбивается при падении с некоторого этажа, то он разбивается и при падении со всех этажей с большим номером.

Разбившиеся транзисторы снова использовать нельзя, а если транзистор остался целым после падения, его можно использовать повторно. Для того, чтобы поднять оставшийся целым транзистор, профессору надо спуститься на первый этаж. Оказавшись на первом этаже, профессор может поднять все лежащие там транзисторы.

Годы профессора уже дают о себе знать, поэтому он хочет минимизировать суммарное расстояние, которое ему придется подниматься по лестнице. Но, возраст дает и определенные преимущества — сняв очки, профессор может с любого этажа определить, разбился транзистор или нет.

Изначально профессор находится на первом этаже, и у него имеется  $m$  транзисторов. В доме, в котором живет профессор,  $n$  этажей.

Найдите минимальное число этажей, которое профессору в худшем случае придется подниматься вверх по лестнице во время проведения экспериментов.

### Формат входного файла

Во входном файле заданы два целых числа — высота дома  $n$  ( $2 \leq n \leq 50$ ) и количество транзисторов  $m$  ( $1 \leq m \leq 10$ ).

### Формат выходного файла

В выходной файл выведите единственное число — минимальное расстояние в этажах, которое в худшем случае придется подниматься вверх по лестнице профессору во время эксперимента.

#### Пример

tester.in	tester.out
5 2	3
2 2	0

В первом из приведенных примеров оптимальное поведение профессора следующее. Сначала следует подняться на два этажа и бросить транзистор с третьего. Если транзистор разобьется, то следует спуститься на второй этаж и попытаться бросить транзистор оттуда — если транзистор разбивается и при бросании со второго этажа, то результат 2, иначе — 3. Если же транзистор не разобьется при падении с третьего этажа, то придется подняться на четвертый и бросить транзистор оттуда. Если он разобьется, то результат 4, если нет — 5. В худшем случае придется подняться на три этажа. Во втором примере ничего бросать не надо, результат исследования — 2.