

## Задача A. Monkeys

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

Учёные поставили эксперимент с обезьяной, обученной колотить одним пальцем по клавишам пишущей машинки. После ряда наблюдений установили, что выбор очередной клавиши не зависит от того, какие клавиши были нажаты перед этим, и для каждой латинской буквы выяснили вероятность того, что в данный момент именно она будет отпечатана. Так как на данном экземпляре пишущей машинки всего 26 клавиш, то сумма этих вероятностей равна 1.

Ваша задача — по имеющимся 26 вероятностям вычислить, какова вероятность того, что первое из двух заданных слов будет напечатано обезьяной ранее второго (слово считается напечатанным, как только оно первый раз в набранной обезьяной строке в качестве подстроки).

### Формат входного файла

В первой строке входного файла заданы 26 неотрицательных чисел, сумма которых равна 1.  $i$ -е число задаёт вероятность того, что при очередном нажатии обезьяна напечатает  $i$ -ю по порядку с начала латинского алфавита букву. Во второй строке через пробел заданы два непустых слова. Слова состоят из строчных латинских букв, длина каждого из слов не превосходит 16. Гарантируется, что вероятность появления каждого из слов больше нуля и что строки подобраны таким образом, что обезьяна не сможет напечатать оба эти слова одновременно (то есть случаи типа `Monkey` и `key` исключены).

### Формат выходного файла

В выходной файл выведите одно число — вероятность того, что первое из заданных слов будет напечатано обезьяной ранее второго, вычисленную с точностью до  $10^{-7}$ .

### Пример

standard input	
0.5 0.5 0	
baba abaa	
standard output	
0.6428571428571429	

## Задача В. Keys

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

В некоторой системе шифрования в качестве ключа можно использовать строки со следующими свойствами:

- ключ состоит только из строчных латинских букв;
- сумма числовых значений букв ключа (то есть порядковых мест каждой из букв в латинском алфавите, например, для `a` числовое значение равно 1, для `z` — 26) должна быть равна 42;
- перестановка двух последних букв ключа даёт строку, в которой буквы идут строго по убыванию своих числовых значений;
- ключ не содержит ни одной из букв `a`, `e`, `i`, `o`, `u` и `y`.

Выведите в лексикографическом порядке все допустимые ключи, имеющие заданную длину.

### Формат входного файла

Во входном файле записано целое число  $L$  ( $2 \leq L \leq 41$ ) — длина каждого из ключей, которые Вам нужно сгенерировать.

### Формат выходного файла

В случае, если допустимые ключи заданной длины существуют, в выходной файл выведите все такие ключи, отсортированные лексикографически. В случае, если не существует ни одного допустимого ключа заданной длины, выведите 0.

## Пример

standard input	standard output
3	pln qkn qlm rgq rhp rjn rkm sfq sgp sjm skl tcs tdr tfp thn tjl vbr vcq vdp vfn vgm vhl wbq wcp wfm wgl whk xbp xdn xfl xgk xhj zbn zcm zdl zfl
41	0

## Задача C. Satellite

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

На «геостационарной» орбите расположены  $K$  спутников связи. Каждому из спутников сопоставлена его координата  $x_i$  — длина участка орбиты от спутника до точки на орбите, находящейся над Центром Управления Полётом, отсчитываемая по часовой стрелке.

В порядке обновления систем спутниковой связи перед запланированной лунной экспедицией Центр планирует запустить  $N$  новых спутников, расположив их на той же орбите так, чтобы минимизировать максимальную длину участка орбиты между двумя соседними спутниками.

По заданным координатам первоначальных спутников, количеству новых спутников и длине орбиты  $O$  вычислите, какой будет максимальная длина участка орбиты между двумя соседними спутниками после обновления системы.

### Формат входного файла

В единственной строке входного файла сначала записаны три целых числа  $K$ ,  $N$  и  $O$  — соответственно первоначальное количество спутников, количество новых спутников и длина орбиты ( $0 \leq K \leq 2 \times 10^4$ ,  $0 < N \leq 2 \times 10^4$ ,  $0 < O \leq 10^7$ ). Далее заданы  $K$  попарно различныз чисел  $x_i$  — координаты существующих спутников ( $0 \leq x_i < O$ ).

### Формат выходного файла

Выведите одно число — максимальную длину участка орбиты между двумя соседними спутниками после обновления системы с точностью до  $10^{-7}$ .

### Пример

standard input	standard output
0 2 10	5
2 3 9 0 4.5	2.25

## Задача D. Landing

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

Для того, чтобы во время полёта на Луну выбросить зонд с аппаратурой, посадочный модуль действует следующим образом: он зависает над поверхностью Луны на некоторой высоте, сбрасывает зонд и затем улетает.

При ударе зонда о поверхность Луны аппаратура остаётся работоспособной, если скорость не превышает заданного значения  $V$ . Конструкция систем управления посадочным модулем такова, что модуль может зависнуть только на высоте, измеряемой целым количеством метров.

Ваша задача — найти максимальную высоту, с которой можно сбросить зонд, чтобы аппаратура не повредилась. Ускорение свободного падения на Луне считать постоянным, не зависящим от высоты и равным 1.6 метров, делённых на секунду в квадрате.

### Формат входного файла

В единственной строке входного файла записано одно целое число  $V$  ( $1 \leq V < 500$ ) — максимальная скорость в метрах в секунду, при которой аппаратура может выдержать удар о поверхность Луны.

### Формат выходного файла

Выведите одно целое число — максимальную высоту в метрах, с которой можно сбросить зонд на поверхность Луны, не повредив аппаратуры.

### Пример

<code>standard input</code>	<code>standard output</code>
4	5
5	7

## Задача E. Мар

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

Вам задана схема некоторого архипелага, представляющая из себя нарисованный на клетчатой бумаге прямоугольник  $L \times K$ . Клетки, соответствующие суше, обозначены точками или цифрами от 1 до 9, клетки, соответствующие океану, обозначен символом W. Назовём две клетки смежными, если они имеют общую сторону.

Для того, чтобы схема была корректна, должны выполняться следующие условия:

- между любыми двумя клетками, обозначающими океан, можно последовательно пройти, всякий раз переходя в смежную клетку, обозначающую океан;
- если между некоторыми двумя клетками, обозначающими сушу, можно последовательно пройти, всякий раз переходя в смежную клетку, обозначающую сушу, то эти две клетки принадлежат к одному и тому же острову;
- на каждом острове есть ровно одна клетка, в которой записана цифра, и число, образуемое этой цифрой, совпадает с количеством клеток, образующих данный остров.
- не должно существовать квадрата  $2 \times 2$ , целиком состоящего из клеток, обозначающих океан.

Проверьте, является ли заданная схема корректной.

### Формат входного файла

В первой строке входного файла записаны два целых положительных числа  $K$  и  $L$ , не превосходящих 50 — количество строк и столбцов карты. Затем в  $K$  последующих строках задана карта. Каждая из строк длины  $L$  представляет очередную строку карты (сверху вниз), при этом символ W обозначает океан, точка или цифра от 1 до 9 — сушу.

### Формат выходного файла

Если карта корректна, выведите YES, в противном случае выведите NO.

### Пример

standard input	standard output
3 3 WW. 2W3 .W.	YES
3 3 WW. .W3 1W.	NO
3 3 .W. WWW 1W1	NO

## Задача F. Easy problem

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 Mebibytes

Эта задача по ошибке попала в набор задач с одного из пробных туров. По заданному  $N$  вычислите  $N$ -е число Фибоначчи. Напомним, что  $F_1 = 1$ ,  $F_2 = 1$ ,  $F_k = F_{k-1} + F_{k-2}$ .

### Формат входного файла

Во входном файле записано целое число  $N$  ( $1 \leq N \leq 10^8$ ) — номер числа Фибоначчи.

### Формат выходного файла

В выходной файл выведите  $N$ -е число Фибоначчи. В случае, если по какой-то причине цифр в этом числе оказалось больше 8, выведите первые 4 цифры числа, затем три точки подряд (они обозначают многоточие), затем последние 4 цифры.

### Пример

standard input	standard output
1	1
3	2
5	5
68	7272...8141