

Задача A. Boolean

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

На Очень Важном Стратегическом Объекте используется программное обеспечение, написанное на Совершенно Секретном Языке Программирования. Про этот язык программирования известно только то, что логические (булевы) выражения в нём устроены следующим образом:

```
выражение ::= переменная
            | '(' выражение ' ' operator ' ' выражение ')'
            | '!' выражение
operator ::= '|' | '&'
переменная ::= 'a' | 'b' | 'c'
```

В связи со введённым на объекте режимом экономии поступил приказ заменить все булевы выражения в программном обеспечении эквивалентными булевыми выражениями, записываемыми минимальным количеством символов, отличных от пробела.

Напишите программу, по заданному булеву выражению вычисляющую минимальное количество символов без учёта пробелов, которое потребуется для записи эквивалентного ему выражения.

Формат входного файла

Во входном файле задана одна непустая строка — булево выражение, записанное по правилам Совершенно Секретного Языка Программирования. Длина строки не превосходит 500 байт.

Формат выходного файла

Выведите одно число — минимальное количество символов, которое потребуется для записи эквивалентного заданному выражения без учёта пробелов.

Пример

standard input	standard output
<code>((a & !b) & (b & !c)) & (c & !a)</code>	6

Задача В. Cryptography

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `64 Mebibytes`
Ограничение по памяти: `6 seconds`

При шифровании внутренней документации Очень Важного Стратегического Объекта используется следующая схема. Пусть требуется зашифровать слово длины N . Тогда используются несколько «ключей» — перестановок. Перестановки задаются первыми N строчными буквами латинского алфавита.

К шифруемому слову применяется последовательно некоторое количество «ключей». Например, пусть заданное слово — `TCMSE`, а «ключи» — `abcd`, `eabcd`, и мы применяем три операции: сначала первый ключ, потом второй, потом снова первый. Тогда получается последовательно `TCMES`, `STCME`, и в итоге — `STCEM`.

Вам нужно решить следующую задачу: по заданному набору ключей и двум словам выяснить, можно ли второе слово получить из первого с помощью ключей из заданного набора не более, чем за L операций. Если можно — выведите наименьшее количество применений ключей, которое переводит первое слово во второе.

Формат входного файла

В первой строке входного файла заданы три целых числа N — длина слова, P — количество ключей и L — максимально разрешённое количество операций ($1 \leq N \leq 26$, $1 \leq P, L \leq 10$). В следующей строке заданы два слова, состоящих из N латинских букв каждое — исходное и зашифрованное. В следующих P строках заданы «ключи» — перестановки из первых N строчных латинских букв.

Формат выходного файла

Выведите одно число — минимальное количество операций, или -1 , если преобразование за не более, чем L операций невозможно

Примеры

standard input	standard output
5 2 4 TCMSE STCEM abcd eabcd	3
5 2 1 TCMSE STCEM abcd eabcd	-1

Задача C. Modules

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

Очень Важный Стратегический Объект состоит из некоторого количества различных модулей, для каждого из которого известна надёжность (ожидаемое количество отказов в год), а также его стоимость. При отказе какого-то модуля он заменяется на новый. Если запасного модуля нужного типа нет, объект выходит из строя.

В связи с финансовым кризисом на поддержание работоспособности Очень Важного Стратегического Объекта в текущем году из бюджета выделена небольшая сумма, которую, к тому же, необходимо израсходовать сразу же после получения, закупив определённое количество запасных модулей разного типа так, чтобы вероятность того, что объект доработает до конца года, была максимальной.

Напомним, что по известной надёжности модуля Q_i вероятность того, что модуль i откажет k раз, равна $P(k) = (e^{-Q_i} \cdot Q_i^k) / k!$.

Ваша задача — найти требуемую максимальную вероятность.

Формат входного файла

В первой строке входного файла заданы 2 целых числа — количество различных модулей N ($1 \leq N \leq 500$) и выделенная сумма P ($0 \leq P \leq 500$). В следующей строке заданы N чисел, заданных с двойной точностью — надёжности модулей. i -е число задаёт надёжность i -го модуля Q_i ($0.0 \leq Q_i \leq 5.0$). В следующей строке заданы n целых чисел P_i . i -е число задаёт стоимость P_i одного i -го модуля, $1 \leq P_i \leq 100$.

Формат выходного файла

Выведите одно число — требуемую вероятность с точностью до 10^{-5} .

Пример

standard input	standard output
2 4 0.6 0.2 4 5	0.71893

Задача D. Passwords

Имя входного файла: **standard input**
Имя выходного файла: **standard output**
Ограничение по времени: **2 seconds**
Ограничение по памяти: **256 Mebibytes**

Согласно инструкции образца 1970 года, пароли доступа к компьютерам Очень Важного Стратегического Объекта должны храниться в особом сейфе зашифрованными лично системным администратором с помощью шифровальной решётки.

Шифровальной решёткой называется бумажный квадрат размера $N \times N$ клеток, в котором вырезаны $\frac{N^2}{4}$ клеток-окошек. Наложив решётку на листок бумаги, имеющий такой же размер, системный администратор пишет в её окошках первые $\frac{N^2}{4}$ символов своего пароля.

После этого он поворачивает решётку по часовой стрелке на 90 градусов. При таком расположении все ранее написанные буквы оказываются под решёткой, а в окошках появляется чистая бумага.

Он записывает в окошках следующие $\frac{N^2}{4}$ символов своего пароля, после чего вновь поворачивает решётку на 90 градусов. Записав очередные $\frac{N^2}{4}$ символов, системный администратор делает ещё один поворот решётки и пишет последние $\frac{N^2}{4}$ символов пароля.

Теперь, не имея такой же шифровальной решётки, очень трудно восстановить пароль по получившемуся квадрату с N^2 символами. Осталось только вырезать окошки таким образом, чтобы при поворотах никакое из вырезанных окошек не накладывалась на одну и ту же клетку дважды. Также нужно учесть, что изначально какие-то окошки уже вырезаны.

Гарантируется, что число вырезанных окошек не больше $\frac{N^2}{4}$ и при поворотах они не будут накладываться на одну и ту же клетку дважды.

Формат входного файла

В первой строке входного файла записано четное целое число N ($1 \leq N \leq 100$) — размер шифровальной решетки. Далее следует N строк, содержащих по N символов. Каждый j -тый символ i -той строки равен 0, если на этом месте в шифровальной решетке вырезано окошко, иначе он равен 1.

Формат выходного файла

В выходной файл следует вывести N строк по N символов в каждой. Каждый j -тый символ i -той строки должен быть равен 0, если на этом месте в шифровальной решетке вырезано окошко. Иначе он должен быть равным 1.

Если возможных решений несколько, выведите любое.

Если решений не существует, выходной файл должен содержать единственную строку 'No solution' без кавычек.

Примеры

standard input	standard output
4	1111
1111	0110
1111	1011
1111	1110
1111	
2	01
11	11
11	

Задача E. Shooting

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

Сотрудники Очень Важного Стратегического Объекта должны регулярно сдавать норматив по стрельбе. Правила сдачи норматива выглядят следующим образом. Если участник попадает в 10-ку, он получает 1 зачётный балл и дополнительный патрон. Если участник попадает в 8-ку или 9-ку, он получает 1 зачётный балл. Попадание в 7-ку и худшие результаты баллов не приносят. Изначально участнику выдаётся k патронов и требуется набрать m зачётных баллов.

Один из сотрудников гарантированно попадает в мишень, но при этом выбивает любой результат от 1 до 10 с вероятностью 0.1. По заданным k и m вычислите вероятность того, что ему удастся сдать норматив.

Формат входного файла

В первой и единственной строке входного файла заданы 2 целых числа: k — количество изначально выданных патронов и m — количество зачётных баллов, которое требуется набрать для сдачи норматива ($1 \leq k \leq 100$, $0 \leq m \leq 100$).

Формат выходного файла

Выведите одно число — требуемую вероятность с точностью до 10^{-3} .

Примеры

standard input	standard output
1 1	0.3
2 10	0.0000

Задача F. Tests

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

На Очень Важном Стратегическом Объекте отбор сотрудников для Стратегического Проекта Галактической Важности производится по следующей схеме. 10 претендентам присвоены номера с 1 до 10. Изначально претенденты выстраиваются в очередь в порядке присвоения номеров. Далее несколько раз повторяется следующее действие: претенденты, стоящие на данный момент на первом и втором месте в очереди, решают на скорость некоторую задачу (задачи каждый раз различны). Победитель остаётся на первом месте в очереди, проигравший отправляется в конец (соответственно, оставшаяся очередь сдвигается на 1 вперёд). По заданным вам позадачным результатам вычислите, в каком порядке будет идти очередь после соответствующего количества итераций.

Формат входного файла

В первой и единственной строке входного файла задано одно число n ($0 \leq n \leq 100$) — количество задач, за которым следуют n символов, задающих результаты для каждой задачи. i -й символ равен 'W', если победу по i -й задаче одержал участник, стоящий на первом месте в очереди, и 'L' в противном случае.

Формат выходного файла

Выведите перестановку чисел от 1 до 10 — порядок, в котором окажутся претенденты в соответствии с показанными ими результатами. Первым выводится участник, стоящий на данный момент в очереди первым.

Примеры

standard input	standard output
0	1 2 3 4 5 6 7 8 9 10
6 WLWWLW	6 8 9 10 2 1 4 5 3 7