

Задача А. Автомат

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче детерминированный автомат — это ориентированный граф, из каждой вершины которого выходит 26 ребер, помеченных строчными латинскими буквами. Обозначим как $d(s, c)$ состояние автомата, в которое ведёт ребро, помеченное символом c , из состояния s . Рассмотрим строку u и состояние автомата s . Определим функцию $T_s(u)$ так:

$$T_s(u) = \begin{cases} s, & \text{если } u \text{ — пустая строка,} \\ d(T_s(u'), c), & \text{если } u = u'c, \text{ где } c \text{ — один символ.} \end{cases}$$

Неформально говоря, $T_s(u)$ — это состояние, в которое мы придём из состояния s по пути, буквы вдоль которого образуют строку u .

Вам дана строка u , состоящая из строчных латинских букв, и детерминированный автомат. Необходимо для каждого состояния автомата s посчитать функцию $T_s(u)$. Строка u может быть очень длинной, поэтому она будет вам дана в закодированном виде.

Строка u будет закодирована алгоритмом RLE. Это значит, что в ней может встречаться выражение вида $(x)k$, где x — тоже строка, закодированная RLE, а k — число (в нашей задаче $2 \leq k \leq 10^9$). При декодировании это выражение преобразуется в строку x , повторённую k раз.

Формальное описание грамматики кодов выглядит так:

$letter := 'a' \mid 'b' \mid 'c' \mid \dots \mid 'z'$

$string := letter \mid string_1 string_2 \mid (string)number$

где $string_1 string_2$ — это конкатенация двух строк, $number$ — это натуральное число без ведущих нулей.

Например, $((ab)2c)2 = (ababc)2 = ababcbabbc$.

Формат входных данных

В первой строке входного файла находится закодированная строка. Её длина — от 1 до 10 000 символов. Эта строка состоит из символов '(', ')', цифр и строчных букв латинского алфавита. Вторая строка содержит натуральное число n — количество состояний автомата ($1 \leq n \leq 1000$). Состояния нумеруются с единицы. Следующие n строк содержат по 26 чисел каждая; i -ая из этих строк содержит значения $d(i, 'a')$, $d(i, 'b')$, ..., $d(i, 'z')$.

Формат выходных данных

Выведите в первую строку n чисел — значения функции T на каждом из состояний.

Пример

стандартный ввод																									
(ab)2b																									
2																									
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
стандартный вывод																									
2 1																									

Задача В. Кустистость деревьев

Имя входного файла: *стандартный ввод*

Имя выходного файла: *стандартный вывод*

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Рассмотрим корневые деревья с n вершинами. Назовем *кустистостью* дерева количество его вершин, у которых более 1 потомка.

Рассмотрим процесс построения корневого дерева, начиная с единственной вершины — корня. В каждый момент времени есть рассматриваемая вершина и очередь необработанных вершин (изначально пустая). К рассматриваемой вершине применяется одно из следующих действий:

- В p процентах случаев у текущей вершины появляется потомок, который добавляется в очередь необработанных вершин.
- В остальных случаях текущая вершина становится обработанной, а текущей становится следующая в очереди вершина.

Исключений из этих правил два:

- Текущая вершина является последней в очереди. В таком случае у неё сразу появляется потомок.
- Дерево уже содержит n вершин. В таком случае процесс заканчивается.

Необходимо найти среднюю кустистость дерева в момент завершения процесса построения.

Формат входных данных

В первой строке находятся два целых числа n и p — количество вершин в дереве и процент случаев, в которых у вершины появляется потомок ($1 \leq n \leq 7500$, $0 \leq p \leq 100$).

Формат выходных данных

Выведите единственное вещественное число — среднюю кустистость дерева с n вершинами. Число выведите с максимально возможной точностью. Ваш ответ будет признан правильным, если его абсолютная погрешность будет не больше 10^{-6} .

Пример

стандартный ввод	стандартный вывод
4 30	0.51

Задача С. Билеты в кино

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В этой задаче вам предлагается помочь кассиру, продающему билеты в кино. Осталось два свободных ряда, один за другим, по m мест в каждом. Места в каждом ряду нумеруются слева направо числами от 1 до m . В очереди стоят люди группами по a_i человек. Каждую группу можно посадить в один из рядов подряд, либо, если a_i чётное, можно посадить её в два ряда на места с одинаковыми номерами.

Кассир в раздумьи: удастся ли ему посадить все группы, соблюдая эти требования? Помогите ему, найдя минимальную длину ряда m , при которой можно посадить все группы, соблюдая их.

Формат входных данных

В первой строке входного файла находится целое число n — количество групп ($1 \leq n \leq 1000$). Во второй строке заданы через пробел n натуральных чисел a_1, a_2, \dots, a_n ; здесь a_i — количество людей в i -й группе. Сумма всех a_i не превосходит 100 000.

Формат выходных данных

Выведите в выходной файл одно целое число — минимальную длину одного ряда, при которой получится посадить все n групп.

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	5
3 12 5 3	11

Задача D. Сортировка

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим перестановку чисел от 1 до n . Будем сортировать её следующим образом: выберем случайно два различных элемента перестановки и поменяем их местами. Будем так делать, пока перестановка не окажется отсортированной.

Вам необходимо подсчитать среднее количество операций при таком странном способе сортировки.

Формат входных данных

В первой строке входного файла находится целое число n — количество элементов перестановки ($1 \leq n \leq 15$). Во второй строке заданы n различных целых чисел от 1 до n — элементы перестановки.

Формат выходных данных

Выведите в выходной файл единственное целое число — искомое среднее количество операций, округлённое вниз.

Замечание

Если ваша программа работает при $n \leq 10$, то она получит не менее 70 баллов.

Пример

стандартный ввод	стандартный вывод
5 1 3 5 4 2	130