

Задача А. Готовь лыжи летом

Имя входного файла: skier.in
Имя выходного файла: skier.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мб
Отображение результатов:

Член сборной по лыжным видам спорта Петя готовится к Олимпийским играм в Ванкувере. Он будет выступать в лыжных гонках на длинные дистанции. Для того, чтобы показывать лучшие результаты на длинных дистанциях, необходимо уметь правильно распределять силы: когда нужно прибавлять, а когда переводить дух. Тем более, если Петя хорошо разогнался, то он может несколько метров не работать, а очень быстро ехать по инерции.

Личный Петин тренер Вася разбил дистанцию на n равных по длине участков. На каждом из участков Петя работает с разной интенсивностью, выраженной целым числом от 1 до n . Таким образом, каждому распределению сил Пети соответствует некоторая перестановка A_k из n чисел. При этом, скорость Пети на каждом отрезке есть максимум из 2 чисел: самого числа A_k (интенсивности работы Пети на k -м отрезке) и его скорости на предыдущем отрезке, уменьшенной на 1 (Петя едет по инерции).

Известно также, что на i -м участке трассы находится опасный поворот, проходить который можно только со скоростью, не превышающей h .

Ваша задача: найти способ наилучшего распределения сил по трассе, то есть такую перестановку A_k , при которой Петя прибудет к финишу за наименьшее время (с учетом вышеобозначенного условия относительно скорости на i -м участке).

Формат входных данных

Во входном файле заданы числа n ($1 \leq n \leq 100\,000$), i ($1 \leq i \leq n$) и h ($1 \leq h \leq n$).

Формат выходных данных

Выведите искомую перестановку A_k .

Примеры

skier.in	skier.out
6 2 3	4 1 6 2 5 3

Задача В. Набор строк

Имя входного файла: `typing.in`
Имя выходного файла: `typing.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов:

В Инновационном Отделе НИИ Исследований Данных Строк разработана клавиатура для внутреннего пользования, облегчающая набор строк огромной длины. Кроме обычных клавиш, соответствующих маленьким латинским буквам, на клавиатуре есть еще n функциональных клавиш F_1, \dots, F_n , соответствующих заданным строкам из словаря S_1, \dots, S_n . При нажатии такой клавиши F_i строка S_i загружается во внутреннюю память клавиатуры. В каждый момент времени в памяти может находиться не более одной строки из словаря.

Кроме того, в клавиатуру встроен графический манипулятор «Кыш», с помощью которого легким движением руки можно ввести любую подстроку находящейся в памяти строки.

Вася занимается исследованием эффективности данного нововведения. Для этого ему требуется написать программу, которая будет вычислять минимальное необходимое количество действий (нажатий и использований «Кыш») для ввода данной строки S . В момент начала ввода строки память пуста.

Например, если требуется ввести строку “abacaba”, а в словаре есть строки “baba” и “caca”, то это можно сделать за четыре действия — нажать F_1 , выбрать манипулятором подстроку “aba”, затем нажать ‘с’, и опять выбрать манипулятором подстроку “aba”. Если бы нужно было набрать с таким словарем “bacababa”, то это можно сделать за пять действий: ‘b’, F_2 , “aca”, F_1 , “baba”.

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 50$). В последующих n строках заданы S_i , составленные из не более чем 500 символов. В последней строке вводится непустая строка S , длина которой не превосходит 100 000. Все символы строк — маленькие латинские буквы.

Формат выходных данных

Выведите минимальное необходимое количество действий.

Пример

typing.in	typing.out
2 baba caca abacaba	4
2 baba caca bacababa	5

Задача С. Округление

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

Заданы n чисел a_1, a_2, \dots, a_n . Все эти числа получены нашими статистиками и описывают количество приверженцев различных вероисповеданий в нашей конторе. К сожалению, босс хочет видеть эти числа не в виде долей единицы и даже не в процентах, а в виде целых долей числа m , где число m — некоторое положительное целое. Потому каждое из этих чисел делится на их сумму, а затем умножается на m . Кроме того, если число после умножения оказалось нецелым, то вы имеете право его округлить до целого в любую сторону. Однако босс поставил вполне логичное условие — сумма получившихся чисел должна быть равна m . Напишите программу, которая делает требуемое округление!

Формат входных данных

В первой строке входного файла записаны два целых числа n и m ($1 \leq n \leq 5000$, $1 \leq m \leq 10000$). В следующей строке записана последовательность n целых неотрицательных чисел a_1, \dots, a_n ($1 \leq \sum a_i \leq 100000$).

Формат выходных данных

В выходной файл необходимо вывести округленную последовательность (n целых чисел через пробел). Сумма получившихся чисел должна быть равна m .

Пример

<code>round.in</code>	<code>round.out</code>
5 4 1 2 3 4 5	0 1 1 1 1

Задача D. Белые прямоугольники

Имя входного файла: `rect.in`
Имя выходного файла: `rect.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт
Отображение результатов:

Петя и Вася играют в игру на доске $m \times n$ клеток. Игра начинается с целиком белой доски. Игроки ходят по очереди. В свой ход игрок может перекрасить прямоугольник, состоящий из белых клеток, в черный цвет. Проигрывает тот, кто не может сделать хода. Петя захотел проанализировать игру. Для этого ему нужно знать, сколько различных ходов существует из той или иной позиции. Напишите программу, подсчитывающую количество возможных ходов.

Формат входных данных

На первой строке входного файла записаны два числа: m и n ($1 \leq m, n \leq 3000$). Последующие m строк задают позицию и содержат по n символов каждая. Белые клетки обозначаются буквой W, а черные — буквой B.

Формат выходных данных

Выходной файл должен содержать одно число — количество различных ходов в данной позиции.

Примеры

<code>rect.in</code>	<code>rect.out</code>
3 3 WWW WBW WWW	20