

Задача А. Беги, Альф! Беги!

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

Прямо сейчас Альфу снится кошмар. В нем он бежит по дороге с препятствиями, на которой, ко всему прочему, разбросаны монеты.

Дорога представляет из себя таблицу $n \times 3$, в клетках которой либо ничего нет, либо находится стена, либо монета. Альф бежит вдоль стороны длиной n . Начинает он бежать из первой строки (то есть у него есть три варианта начала, он может выбрать любой из них) и бежит до тех пор, пока не врежется в стену, либо не пробежит дорогу целиком (не окажется в строчке n).

Пусть сейчас Альф стоит в строке x и столбце y — $(x; y)$, тогда он может попасть в три возможные клетки: $(x + 1; y - 1)$, $(x + 1; y)$, $(x + 1; y + 1)$, если конечно новая клетка не выходит за пределы дороги, и в ней не находится стена. Так как все обитатели планеты Мелмак умеют контролировать свои сны, Альф смог получить карту дороги. Теперь он хочет узнать, какое наибольшее количество монет можно собрать к концу забега.

Так как контроль сна отнимает у Альфа много сил, он просит вас написать программу, которая по карте сможет определить наибольшее количество монет, которое можно собрать за один забег.

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 10^4$) — количество строк в таблице. В следующих n строках дано по три символа, характеризующие данную строку таблицы. \cdot равен «.», если клетка пустая, «С», если в этой клетке монета, и «W», если стена. Если в первой строке во всех клетках находятся стены, Альф заканчивает забег сразу.

Формат выходных данных

В выходной файл выведите одно число — наибольшее количество монеток, которые можно собрать.

Примеры

стандартный ввод	стандартный вывод
5 W.W C.C WW. CC. CWW	3
4 W.W CWC W.W CWW	2

Задача В. Черепаха и монеты

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

Черепаха хочет переползти из левого верхнего угла поля размером n на m клеток ($2 \leq n, m \leq 1000$) в правый нижний. За один шаг она может переместиться на соседнюю клетку вправо или на соседнюю клетку вниз. Кроме того, проходя через каждую клетку, Черепаха получает (или теряет) несколько золотых монет (это число известно для каждой клетки).

Определите, какое максимальное количество монет может собрать Черепаха по пути и как ей нужно идти для этого.

Формат входных данных

В первой строке вводятся два натуральных числа: n и m ($2 \leq n, m \leq 1000$), разделённые пробелом. В каждой из следующих n строк записаны через пробел по m чисел a_{ij} ($|a_{ij}| \leq 10$), которые обозначают количество монет, получаемых Черепашкой при проходе через каждую клетку. Если это число отрицательное, Черепашка теряет монеты.

Формат выходных данных

В первой строке программа должна вывести наибольшее количество монет, которое может собрать Черепаха. Во второй строке без пробелов выводятся команды, которые нужно выполнить Черепашке: буква 'R' (от слова **right**) обозначает шаг вправо, а буква 'D' (от слова **down**) — шаг вниз.

Примеры

стандартный ввод	стандартный вывод
3 3 0 2 -3 2 -5 7 1 2 0	6 RRDD
4 5 4 5 3 2 9 4 6 7 5 9 5 2 5 -3 -10 3 5 2 9 3	41 RDRDDRR

Задача С. Строй штурмовиков

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

На планете Татуин снова проходит сражение между армией штурмовиков и повстанцев. Армия штурмовиков представляет из себя укомплектованный строй из $n \times m$ солдат. После продолжительной битвы многие солдаты пали и войска штурмовиков поредел. Максимилиан Вирс всегда отличался умением вести битвы, и на этот раз у него есть разрушительная стратегия, но для ее исполнения ему необходимо узнать обстановку в его войске.

Он отправляет одного из командеров штурмовиков сесть на гравцикл и пролететь над войском для его оценки. Если точнее, ему необходимо знать, какой максимальный квадрат его армии остался уцелевшим. Командер видит строй и может сказать о том, в какой точке присутствует штурмовик, а в какой — уже пал. Помогите командеру найти длину стороны такого максимального квадрата, где войско осталось целым по данным, которые он вам предоставит.

Формат входных данных

В первой строке даны два натуральных числа n и m ($1 \leq n, m \leq 1000$). Далее в n строках расположено по m чисел, разделенных пробелами (число равно 0, если в позиции отсутствует штурмовик, или 1, если уцелел).

Формат выходных данных

Выведите одно число — сторону максимального квадрата, полностью состоящего из уцелевших штурмовиков.

Пример

стандартный ввод	стандартный вывод
4 5 0 0 0 1 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0	2

Задача D. Рюкзак

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

Дано n предметов массой m_1, \dots, m_n и стоимостью c_1, \dots, c_n соответственно.

Ими наполняют рюкзак, который выдерживает вес не более m . Определите набор предметов, который можно унести в рюкзаке, имеющий наибольшую стоимость.

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 1 000 и натуральное число m , не превышающее 1 000.

Во второй строке вводятся n натуральных чисел m_i , не превышающих 100.

Во третьей строке вводятся n натуральных чисел c_i , не превышающих 100.

Формат выходных данных

В первой строке выведите количество предметов, которые нужно взять. Во второй строке выведите номера предметов (числа от 1 до n), которые войдут в рюкзак наибольшей стоимости.

Пример

стандартный ввод	стандартный вывод
4 6 2 4 1 2 7 2 5 1	3 1 3 4

Задача Е. Расстояние по Левенштейну

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

Дана текстовая строка. С ней можно выполнять следующие операции:

1. Заменить один символ строки на другой символ.
2. Удалить один произвольный символ.
3. Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «СУК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК».

Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна.

Определите расстояние Левенштейна для двух данных строк.

Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

Формат выходных данных

Требуется вывести одно число — расстояние Левенштейна для данных строк.

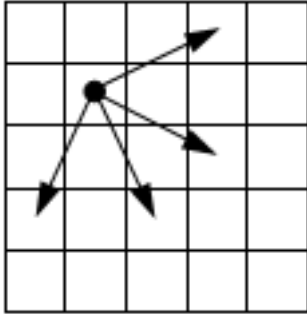
Пример

стандартный ввод	стандартный вывод
ABCDEFGH ACDEXGHI	3

Задача F. Лошадью ходи - 2!

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Шахматный конь должен пройти по клетчатой доске размером n на m клеток из левого верхнего угла в правый нижний. Из каждой клетки есть четыре возможных хода:



Определите, сколькими способами конь может пройти этот маршрут. Поскольку это число может быть очень велико, вычислите его по модулю $10^6 + 7$, то есть найдите остаток от деления этого числа на $10^6 + 7$.

Формат входных данных

Входная строка содержит два натуральных числа: размеры доски n и m ($1 \leq n, m \leq 500$).

Формат выходных данных

Программа должна вывести число возможных маршрутов шахматного коня из левого верхнего угла поля в правый нижний, вычисленное по модулю $10^6 + 7$.

Пример

стандартный ввод	стандартный вывод
5 5	4

Задача G. Невозрастающая подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е. такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длиной $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

стандартный ввод	стандартный вывод
5	3
5	3 4 5
8	
10	
4	
1	

Задача Н. Удаление скобок 2.0

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

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходных данных

Выведите строку максимальной длины, являющейся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.

Пример

стандартный ввод	стандартный вывод
([)]	[]

Задача I. Оптимальное бинарное дерево поиска

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

Рассмотрим множество $S = e_1, e_2, \dots, e_n$, состоящее из n различных элементов таких, что $e_1 < e_2 < \dots < e_n$. Рассмотрим бинарное дерево поиска, состоящее из элементов S . Чем чаще производится запрос к элементу, тем ближе он должен располагаться к корню.

Стоимостью $cost$ доступа к элементу e_i из S в дереве будем называть значение $cost(e_i)$, равное числу ребер на пути, который соединяет корень с вершиной, содержащей элемент. Имея частоту запросов к элементам из S , $(f(e_1), f(e_2), \dots, f(e_n))$, определим общую стоимость дерева следующим образом:

$$f(e_1) \cdot cost(e_1) + f(e_2) \cdot cost(e_2) + \dots + f(e_n) \cdot cost(e_n)$$

Дерево, имеющее наименьшую стоимость, считается наилучшим для поиска элементов из S . Именно поэтому оно называется Оптимальным Бинарным Деревом Поиска.

Формат входных данных

Состоит из нескольких тестов, каждый из которых расположен в отдельной строке.

Первое число в строке n ($1 \leq n \leq 250$) указывает на размер множества S . Следующие n неотрицательных целых чисел описывают частоты запросов элементов из S : $f(e_1), f(e_2), \dots, f(e_n)$. Известно, что $0 \leq f(e_i) \leq 100$.

Сумма n по всем тестам не больше 5000.

Формат выходных данных

Для каждого теста в отдельной строке вывести стоимость Оптимального Бинарного Деревя Поиска.

Пример

стандартный ввод	стандартный вывод
1 5	0
3 10 10 10	20
3 5 10 20	20

Задача J. Дубы

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

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведенные слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
5	2
3 2 4 8 5	4
	2

Задача К. Покраска забора

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

Мэр города Берендея решил построить перед своим домом забор из N деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время.

Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 300$), где N количество досок в заборе. Вторая строка содержит строку из N символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите M — наименьшее возможное время покраски забора в часах. Следующие M строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа L_i и R_i , а также заглавную букву латинского алфавита, задающую цвет C_i и означает, что робот должен покрасить участок забора с L_i по R_i доску в цвет C_i (если длина забора N , должно выполняться $1 \leq L_i \leq R_i \leq N$).

Примеры

стандартный ввод	стандартный вывод
1 A	1 1 1 A
2 AA	1 1 2 A