

Задача А. Гарри Поттер и нос Волан-де-Морта

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

Во времена обучения в Хогвартсе, когда лорда Волан-Де-Морта еще звали Томом Марволо Реддлом, он был довольно привлекательным молодым человеком. Однако многочисленные темные дела испортили его внешний вид, а именно — лишили носа.

Медицина не стоит на месте, и чудеса современной пластической хирургии помогут ему решить эту проблему. Он записался на пластическую операцию по созданию искусственного носа, и теперь перед ним осталась только одна проблема — ему необходимо выбрать место расположения своего будущего носа. К решению этого вопроса он подошел с математической точки зрения.

Лицо лорда можно упрощенно представить как клетчатый прямоугольник из n строк и m столбцов. Волан-де-Морт знает, что его нос займет ровно две клетки, имеющих общую горизонтальную или вертикальную сторону. Еще он знает, что некоторые клетки уже заняты его глазами и ртом, и они не могут быть заняты еще и носом. Теперь он хочет знать количество способов разместить новый нос на своем лице. Чтобы не стать следующей жертвой темных дел Волан-де-Морта, вам придется помочь ему с решением этой задачи.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 1000$) — размеры лица лорда Волан-де-Морта. Каждая из следующих n строк содержит по m символов — описание его лица. Символ «#» означает, что соответствующая клетка уже чем-то занята, а символ «.» — что она свободна и может стать одной из двух, занятых носом.

Формат выходного файла

В первой строке выходного файла выведите одно целое число — количество способов разместить нос на лице Волан-де-Морта.

Примеры

nose.in	nose.out
2 3	7
3 4 ..#. #..# ..#.	5
5 8###.##. #.# .#####.	34

Комментарий

Решения, работающие для лица, на котором нет занятых клеток, будут оцениваться из 40 баллов.

Задача В. Завоеватель

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

И я видел, что Агнец снял первую из семи печатей, и я услышал одно из четырёх животных, говорящее как бы громовым голосом: иди и смотри. Я взглянул, и вот, конь белый, и на нем всадник, имеющий лук, и дан был ему венец;
и вышел он как победоносный, и чтобы победить.

Откровение Иоанна Богослова

Первый всадник Апокалипсиса Завоеватель пришел на Землю. Увидели это жители одного города и решили предупредить свою столицу, дабы ее жители успели покаяться. С этой целью жителями города в столицу был отправлен гонец на лошади.

Дорога между этим городом и столицей представляет собой прямую, на которой, включая этот город и столицу, расположены n городов, причем расстояние между любыми соседними городами на этой прямой одинаково. В каждом из $n - 2$ городов, мимо которых гонцу необходимо проехать, он может сменить лошадь в конюшне. Хозяин каждой конюшни знает, сколько минут требуется лошади в этой конюшне на дорогу между двумя соседними городами. За какое минимальное время гонец сможет добраться до столицы (n -го города на этой прямой)?

Формат входного файла

В первой строке входного файла содержится одно целое число n ($1 \leq n \leq 100\,000$) — количество городов. В следующей строке содержатся n натуральных чисел t_i ($1 \leq t_i \leq 10^6$) — количество минут, необходимое лошади из конюшни в i -ом городе на преодоление расстояния между двумя городами. Заметим, что первое число означает скорость лошади, которая была у гонца при выезде из первого города, а последнее — скорость лошади в конюшне столицы.

Формат выходного файла

В выходной файл выведите одно число — количество минут, за которое гонец доедет до столицы.

Примеры

<code>conquest.in</code>	<code>conquest.out</code>
6 3 4 3 2 1 5	12

Комментарий

Решения, работающие в случаях, в которых n не превышает 100, будут оцениваться в 60 баллов.

Задача С. Держать строй

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

Ник Фьюри решил, что бойцы отряда спецназа, являющегося подразделением организации S.H.I.E.L.D., помогут мстителям отразить атаку войска Локи. Он решил, что в бой отправятся n бойцов, а все остальные понадобятся в других местах. Теперь ему осталось только выбрать, какие именно бойцы пойдут в атаку.

Сначала Ник выбрал n бойцов случайным образом и выстроил их в линию, а затем стал по одному заменять кого-то из уже выбранных бойцов на другого солдата, который в настоящее время в строю не стоит. Поскольку отряд достаточно большой, Ник не знает каждого бойца лично. Оценить боеспособность отряда он может разве что по каким-нибудь заметным внешним признакам. Важным показателем боеспособности отряда является, например, то, стоят ли солдаты в строю по неубыванию роста.

Так, Ник может давать команды двух видов. Первая команда заключается в том, что новый солдат роста x встает в строй вместо солдата, стоящего на k -ом месте. Подавая вторую команду, он хочет узнать, стоят ли солдаты в строю по неубыванию роста. Ваша задача обрабатывать эти команды и сообщать в ответ на запросы то, что хочет узнать Ник.

Формат входного файла

Первая строка входного файла содержит два числа n и m ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$) — количество солдат в строю и количество команд, которые подаст Ник. Вторая строка содержит n целых неотрицательных чисел, не превосходящих 10^9 — исходный рост солдат в строю.

Следующие m строк содержат команды, подаваемые Ником. Если первый символ в строке, описывающей очередную команду, `!`, то за ним следуют два числа k и x ($1 \leq k \leq n$, $0 \leq x \leq 10^9$), где k — место в строю того солдата, которого должен заменить солдат роста x . Команда второго типа описывается знаком `?`.

Формат выходного файла

Для каждой команды второго типа в отдельной строке выведите `YES`, если в данный момент солдаты в строю стоят по неубыванию роста, и `NO` — в противном случае.

Примеры

army.in	army.out
5 5	YES
2 4 6 8 10	NO
?	YES
! 2 7	
?	
! 3 8	
?	

Комментарий

Решения, работающие в случаях, в которых n и m не превышает 10 000, будут оцениваться в 60 баллов.

Задача D. Поручения

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

Учитель Сплинтер всегда держит своих учеников в тонусе. Он дал им n поручений: им нужно помочь Кейси Джонсу в уличной драке, спасти Землю от нападок Шреддера и сорвать коварные планы Кренга, а в довесок еще сделать кучу дел по дому. Причем черепашкам необходимо выполнить все эти задания.

Очевидно, каждое поручение — не из приятных и доставляет черепашкам какое-то количество боли и страданий. Когда черепашки выполняют очередное задание, боль, которую оно приносит, может добавиться к усталости черепашек. Однако, это происходит только в том случае, если любое из заданий, выполненных ими раньше, приносило им меньше боли, чем последнее выполненное. Страдание добавляется к усталости по таким же правилам.

Теперь черепахи хотят выполнять задания в таком порядке, чтобы после выполнения всех заданий усталость была минимальна.

Формат входного файла

В первой строке входного файла задано одно целое число n — количество заданий, которые получили черепашки. ($1 \leq n \leq 10^5$). Далее следуют n строк, где для каждого i -го задания задано два целых числа — количество боли a_i и страдания b_i ($1 \leq a_i, b_i \leq 10^9$). Гарантируется, что все a_i различны и все b_i различны.

Формат выходного файла

В первую строку выходного файла выведите минимальную усталость черепашек после выполнения всех заданий. Во второй строчке выходного файла выведите перестановку чисел от 1 до n — порядок, в котором следует выполнять задания. Если существует несколько оптимальных ответов, выведите любой.

Примеры

<code>missions.in</code>	<code>missions.out</code>
3	8
3 2	2 1 3
2 3	
1 1	

Комментарий

Решения, работающие в случае, когда $n \leq 10$, будут оцениваться в 30 баллов.

Задача Е. Голод

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

И когда Он снял третью печать, я слышал третье животное, говорящее:
иди и смотри.
Я взглянул, и вот, конь вороной, и на нем всадник, имеющий меру в руке
своей.
И слышал я голос посреди четырех животных, говорящий: хиникс
пшеницы за динарий, и три хиникса ячменя за динарий;
елея же и вина не повреждай.

Откровение Иоанна Богослова

До конца света осталось не так много дней, но Третий всадник Апокалипсиса Голод так и не смог понять, какие города стоит наказывать за прегрешения, а какие нет. Тогда он решил потренироваться на одной небольшой стране.

Эта страна представляет собой клетчатый прямоугольник размера $n \times m$, в котором каждая клетка — отдельный город. Голод может либо насладиться голодом на город, либо пощадить его. При этом есть города, в которых еды хватит и до следующего конца света, и Голод не в силах заставить их голодать, а есть те, в которых еды никогда и не было, и им в любом случае придется голодать.

Страшный Суд Голода должен быть еще и справедлив, а это значит, что в любом квадрате размера 2×2 должно быть поровну голодающих и сытых городов. Теперь Голод хочет узнать количество различных вариантов распределения городов этой страны на голодающие и сытые.

Формат входного файла

В первой строке входного файла задано два целых числа n и m ($2 \leq n, m \leq 10^3$) — размер страны. Далее следует n строк по m символов в каждой, где символ '+' означает, что соответствующий город не может голодать, символ '-' — соответствующий город будет голодать в любом случае и символ '.' — Голод может решить, что ему делать с этим городом.

Формат выходного файла

Выведите количество различных вариантов насладиться голодом на эту страну по модулю $10^9 + 7$

Примеры

famine.in	famine.out
3 2 +. .. .-	2

Комментарий

Решения, работающие в случаях, когда $n \times m \leq 20$, будут оцениваться в 10 баллов.

Решения, работающие в случаях, в которых $n \leq 10$ и $m \leq 50$, будут оцениваться в 40 баллов.

Решения, работающие в случаях, в которых $n \leq 15$ и $m \leq 100$, будут оцениваться в 60 баллов.

Задача F. Тессеракт

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

Коварный Локи напал на Землю, используя Тессеракт для того, чтобы переместиться из Асгарда. Понимая ценность этого артефакта, он решает защитить его от использования кем-либо, кроме себя. Для этого он решает немного изменить его внутреннюю структуру так, чтобы только он знал, как её можно восстановить.

После внимательного исследования, внимание Локи привлекла цепочка маленьких кубиков внутри Тессеракта. Среди этих кубиков были крайне похожие друг на друга, что идеально подходило для небольшого, но важного для работы изменения Тессеракта. Проявив восхваленную в легендах коварность, Локи, в надежде на невнимательность людей, решил внести два изменения таких, чтобы после каждого из них, артефакт выглядел бы таким же, как и раньше.

Всего цепочка, которую собирается менять Локи, состоит из n кубиков. Среди них есть похожие, на чём и собирается сыграть Локи. Он берёт какой-то отрезок кубиков в этой цепочке, вырывает его, разворачивает и вставляет обратно. При этом, он выбирает отрезок таким образом, чтобы внешне цепочка не изменилась. Затем он повторяет то же самое с другим отрезком, который содержал в себе первый, но не совпадал с ним. Очевидно, после этих двух операций цепочка уже не будет совпадать с исходной, и артефакт будет испорчен.

Чтобы оценить вероятность быть уличённым в порче Тессеракта, Локи решил выяснить, сколькими способами он мог выбрать первый отрезок.

Рассмотрим, к примеру, цепочку `aabaa`, в которой одинаковыми буквами обозначены похожие кубики. Тогда настоящий Тессеракт содержит цепочку $a_1a_2ba_3a_4$. Локи может, например, проделать следующую последовательность действий: $a_1a_2ba_3a_4 \rightarrow a_2a_1ba_3a_4 \rightarrow a_4a_3ba_1a_2$. Внешне ничего не изменилось, однако цепочка уже другая.

Формат входного файла

В первой и единственной строке задана цепочка, состоящая из маленьких латинских букв, длиной не более 100 000.

Формат выходного файла

Единственное число — количество различных первых действий Локи.

Примеры

<code>tesseract.in</code>	<code>tesseract.out</code>
<code>aabaa</code>	8

Комментарий

Решения, работающие для строк длины не более 100, будут оцениваться в 30 баллов.

Решения, работающие для строк длины не более 4000, будут оцениваться в 60 баллов.