Задача A. 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 ($1 \le n \le 100\,000$) — количество городов. В следующей строке содержатся n натуральных чисел t_i ($1 \le t_i \le 10^6$) — 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

conquest.in	conquest.out
6 3 4 3 2 1 5	12

Пояснение

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

Задача B. Война

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

И когда он снял вторую печать, я слышал второе животное, говорящее:
иди и смотри.
И вышел другой конь, рыжий;
и сидящему на нем дано взять мир с земли, и чтобы убивали друг друга;
и дан ему большой меч.

Откровение Иоанна Богослова

Мир перенаселён. Ресурсы планеты на исходе. Жители всех стран на грани отчаяния. Политическая обстановка накаляется с каждым днём. У государств нет ни сил, ни ресурсов на войну друг с другом. В любой момент в любой стране может вспыхнуть кровопролитная гражданская война. Миллионы жертв неизбежны. В результате прошлых войн многие государства оказались разделены географически, образовались анклавы.

Представим карту мира как прямую, на которой расположены города, принадлежащие разным странам. Если на этой прямой подряд находятся несколько городов, принадлежащих одной стране, то все эти города считаются отдельным анклавом этой страны. Если в каком-то анклаве какой-то страны начинается гражданская война, то этот анклав мгновенно слабеет, после чего его целиком захватывает один из соседних с ним анклавов. Если оба соседних анклава принадлежат одной и той же стране, то они объединяются в один анклав.

Второй всадник Апокалипсиса Война, видя приближение конца света, хочет закончить всё это побыстрее. Он может инкогнито внедрить провокатора в любой анклав любой страны и начать там гражданскую войну, что приведет к исчезновению этого анклава с карты мира. Выясните, за сколько внедрений Война сможет оставить в мире ровно один анклав и начать в нем гражданскую войну.

Формат входного файла

В первой строке задано единственное число n ($1 \le n \le 500$) — количество стран на карте мира. Во второй строке задана сама карта мира: номера стран a_i, которым принадлежат соответствующие города на прямой. Все номера стран являются натуральными числами, не превосходящими 10^9.

Формат выходного файла

Выведите искомое минимальное число внедрений.

Примеры

war.in	war.out
4 1 2 3 1	3
3 1 1 1	1
4 1 2 2 1	2

Пояснение

Решения, работающие при $n \le 10$ и $a_i \le 10$, будут оцениваться в 10 баллов.

Решения, работающие при $n \le 50$ и $a_i \le 1000$, будут оцениваться в 40 баллов.

Задача C. 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 наслать голод 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 ($2 \le n, m \le 10^3$) — 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 наслать голод на эту страну по модулю $10^9 + 7$

Примеры

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

Пояснение

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

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

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

Задача D. Смерть

Имя входного файла:	`death.in`
Имя выходного файла:	`death.out`
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

И когда Он снял четвертую печать, я слышал голос четвертого животного, говорящий: иди и смотри.
И я взглянул, и вот, конь бледный, и на нем всадник, которому имя «смерть»;
и ад следовал за ним;
и дана ему власть над четвертою частью земли — умерщвлять мечом и голодом, и мором и зверями земными.

Откровение Иоанна Богослова

Смерть — Четвертый всадник Апокалипсиса, и за этим всадником следует ад. Однако, даже этот всадник готов пощадить некоторые города и оставить их жителей в живых.

Карта страны, которую изучает Смерть, представляет собой клетчатый прямоугольник размера $n \times m$. Каждая клетка — город, и в каждом городе живут люди, подчиняющиеся одному определенному лорду. Смерть хочет пощадить несколько городов так, чтобы выполнялись два правила:

- из любого выжившего города можно добраться в любой другой, перемещаясь только по выжившим городам, соседним по стороне
- в выживших городах проживают люди, подчиняющиеся не более, чем двум различным лордам

Теперь Смерть заинтересовало максимальное количество городов, которые он может пощадить.

Формат входного файла

В первой строке входного файла задано два целых числа n и m ($1 \le n, m \le 10^3$) — размеры страны. Следующие n строк содержат по m чисел каждая — номера лордов, которым подчиняются люди в соответствующих городах. Номера лордов — натуральные числа, не превышающие 10^6.

Формат выходного файла

Выведите в первой строке одно натуральное число — максимальное количество городов, которые смогут выжить после набега смерти. Во второй — два числа: номера лордов, которым подчиняются люди в выживших городах. В случае, если выживут только люди, подчиняющиеся одному лорду, номер этого лорда должен быть выведен два раза.

Примеры

death.in	death.out
5 5 1 1 2 2 1 1 2 2 1 1 1 2 3 3 1 1 3 3 2 4 1 2 2 4 1	17 1 2

Пояснение

Решения, работающие в случаях, в которых n и m не превышают 20, будут оцениваться в 30 баллов.

Решения, работающие в случаях, в которых n и m не превышают 80, будут оцениваться в 60 баллов.