

Задача А. Высокий маразм

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Город Нью-Васюки представляет собой таблицу N на N кварталов. Таким образом, в нем существует $(N + 1)(N + 1)$ перекрестков и $2N(N + 1)$ двусторонних дорог. Каждый перекресток имеет высоту. Известно, что левый верхний перекресток имеет высоту 0, а правый нижний – высоту 1. Для каждой дороги известно, сколько человек идет в каждом направлении по этой дороге. При этом, если дорога ведет от перекрестка i к перекрестку j , и разность высот $h = h_j - h_i$, то неудобство перемещения для каждого человека равно $\max(h, 0)$.

Для всех клеток, кроме двух угловых вы вправе выбирать любую высоту. Найдите распределение, при котором суммарное неудобство будет минимальным.

Формат входных данных

В первой строке входного файла содержится число N – размер таблицы, $1 \leq N \leq 500$. Далее следует $4N(N+1)$ чисел – искомое число людей для каждой дороги в естественном порядке. Сначала следует $N(N + 1)$ чисел для каждой дороги с запада на восток, затем столько же чисел с севера на юг, потом столько же чисел с востока на запад, а затем столько же чисел с юга на север. Для каждого направления дороги перечислены с севера на юг, а затем – с запада на восток. Обратите внимание на пример (там все числа специально различны).

Все количества являются целыми неотрицательными числами не превосходящими миллиона.

Формат выходных данных

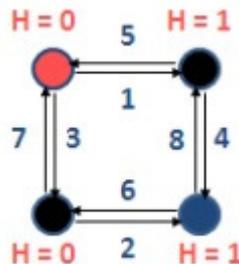
Выведите одно число – итоговое минимальное суммарное неудобство.

Примеры

stdin	stdout
1	3
1	
2	
3	
4	
5	
6	
7	
8	

Замечание

Высоты могут быть произвольными вещественными числами! Картинка к примеру из условия:



Задача В. Мадока и первая сессия

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

О нет, на первой же сессии Мадоке попался билет со следующей сложной задачей:

Дано число n и m пар чисел (v_i, u_i) . А также есть массив b_1, b_2, \dots, b_n , изначально заполненный нулями.

Затем для каждого индекса i , где $1 \leq i \leq m$, выполняется либо $b_{v_i} := b_{v_i} - 1$ и $b_{u_i} := b_{u_i} + 1$, либо $b_{v_i} := b_{v_i} + 1$ и $b_{u_i} := b_{u_i} - 1$. Обратите внимание, что ровно одна из этих операций должна быть выполнена для каждого i .

Также дан массив s размера n , состоящий только из 0 и 1. И массив a_1, a_2, \dots, a_n , где гарантируется, что если $s_i = 0$, то $a_i = 0$.

Помогите Мадоке и определите, можно ли выполнить операции выше таким образом, чтобы для каждого i , где $s_i = 1$, выполнялось $a_i = b_i$. И если возможно, то как это сделать.

Формат входных данных

Первая строка содержит два целых числа n и m ($2 \leq n \leq 10000, 1 \leq m \leq 10000$) — длина массива a и количество пар чисел.

Вторая строка содержит n целых чисел s_1, s_2, \dots, s_n ($0 \leq s_i \leq 1$) — элементы массива s .

Третья строка содержит n целых чисел a_1, a_2, \dots, a_n ($|a_i| \leq m$) — элементы массива a . Гарантируется, что если $s_i = 0$, то $a_i = 0$.

i -я из следующих m строк содержат два целых числа v_i и u_i ($1 \leq v_i, u_i \leq n, v_i \neq u_i$) — индексы элементов массива b , к которым применяется операция. Также гарантируется, что не существует таких двух индексов i и j , где $1 \leq i < j \leq m$, что $(v_i, u_i) = (v_j, u_j)$ или $(v_i, u_i) = (u_j, v_j)$.

Формат выходных данных

Выведите в первой строке «YES», если можно выполнить операции нужным образом, и «NO» в противном случае.

Вы можете выводить каждую букву в любом регистре (например, «YES», «Yes», «yes», «yEs» будут распознаны как положительный ответ).

В случае, если вы вывели «YES», выведите m пар целых чисел. Если для пары (v_i, u_i) нужно выполнить $b_{v_i} := b_{v_i} - 1$ и $b_{u_i} := b_{u_i} + 1$, выведите (v_i, u_i) . Иначе выведите (u_i, v_i) . Если существует несколько способов получить правильный ответ, можно вывести любой из них.

Пары можно выводить в любом порядке.

Примеры

стандартный ввод	стандартный вывод
5 5 1 1 1 1 1 -2 0 2 1 -1 1 5 1 4 3 5 3 4 4 5	YES 1 5 1 4 5 3 4 3 5 4
5 5 0 1 0 1 0 0 1 0 0 0 1 3 2 3 3 5 3 4 4 5	YES 3 1 3 2 5 3 3 4 4 5
4 4 1 1 1 1 0 2 -2 2 1 3 1 4 2 3 2 4	NO

Замечание

В первом примере массив b будет меняться следующим образом:
 $[0, 0, 0, 0, 0] \rightarrow [-1, 0, 0, 1, 0] \rightarrow [-2, 0, 0, 1, 1] \rightarrow [-2, 0, 1, 0, 1] \rightarrow [-2, 0, 2, 0, 0] \rightarrow [-2, 0, 2, 1, -1]$.
 $a_i = b_i$ для всех индексов i от 1 до 5.

Во втором примере нам достаточно, чтобы в конце $b_2 = 1$, поскольку только $s_2 = 1$.

В третьем примере входных данных нельзя выполнить операции нужным образом.

Задача С. Командная олимпиада

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

Совсем скоро начнется первый тур очередной всероссийской командной олимпиады школьников по палеонтологии (ВКОШП). На олимпиаду приехали команды из n школ, от каждой школы приехало ровно по две команды. Команды уже заняли свои места, когда обнаружилось, что некоторые команды из одной школы сидят слишком близко. Перед организаторами олимпиады встала серьезная задача — пересадить участников олимпиады.

Стол, за которыми сидят команды, расставлены в один ряд. Расстояние между рабочими местами соседних команд оказалось равно 10 метрам. Организаторы хотят, чтобы минимальное расстояние между двумя командами из одной школы было как можно больше.

Пересаживая команду, организатором необходимо перенести на новое место все оборудование, приготовленное командой для исследований. Поэтому организаторы хотят пересадить команды так, чтобы сумма расстояний между старыми и новыми рабочими местами команд была как можно меньше.

Например, пусть в соревновании принимают участие по две команды школ 1, 2, 3 и 4. Пусть исходно команды распределены по рабочим местам следующим образом: 1, 3, 2, 2, 1, 4, 4, 3 (для каждой команды указан номер школы, которую она представляет). При таком распределении по рабочим местам команды из школы 2 сидят слишком близко, как и команды из школы 4. Пересадив команды в следующем порядке: 1, 3, 2, 4, 1, 3, 2, 4, жюри может добиться своего: команды из одной школы сидят на местах, расстояние между которыми не меньше 40 м, большего расстояния добиться нельзя. Сумма расстояний между старыми и новыми позициями для данного примера равна $0 + 0 + 0 + 20 + 0 + 20 + 30 + 10 = 80$ м, для исходного распределения команд она минимальна.

Вам задано исходное распределение команд по рабочим местам. Требуется пересадить их так, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми местами рабочими команд была минимально возможной.

Формат входных данных

В первой строке входного файла задано число n — количество команд ($1 \leq n \leq 100$). Во второй строке задано исходное распределение команд по рабочим местам — последовательность a_1, a_2, \dots, a_{2n} . Для каждой команды указан номер школы, которую она представляет. Гарантируется, что последовательность состоит из чисел от одного до n и каждое число встречается ровно два раза.

Формат выходных данных

В единственную строку выходного файла выведите, каким образом следует пересадить команды, чтобы минимальное расстояние между командами из одной школы было как можно больше. При этом среди возможных новых размещений команд следует выбрать такое, чтобы сумма расстояний между старыми и новыми рабочими местами команд была минимально возможной. Если оптимальных ответов несколько, можно вывести любой из них.

Пример

стандартный ввод	стандартный вывод
4 1 3 2 2 1 4 4 3	1 2 4 3 1 2 4 3

Задача D. Кирпичи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Кирпич — прямоугольник с целыми сторонами шириной 1 или высотой 1 (или и то и другое).

Дана сетка $n \times m$, и каждая ячейка окрашена в черный или белый цвет. Замощение — это способ поместить кирпичи на сетку так, чтобы каждая черная ячейка была покрыта **ровно одним** кирпичом, а каждая белая ячейка не была покрыта кирпичом. Другими словами, кирпичи размещаются только в черных ячейках, покрывают все черные ячейки, и **никакие два кирпича не перекрываются**.

Какое минимальное количество кирпичей необходимо для замощения данной сетки?

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leq n, m \leq 200$) — количество строк и столбцов соответственно.

Следующие n строки описывают сетку. i -я строка содержит строку длиной m , где j -я строка обозначает цвет ячейки в строке i , столбец j . Черная ячейка обозначается символом «#», а белая — символом «.».

Гарантируется, что есть хотя бы одна черная ячейка.

Формат выходных данных

Выведите единственное целое число — минимальное количество требуемых кирпичей.

Примеры

стандартный ввод	стандартный вывод
3 4 # . ## #### ## . .	4
6 6 ##### ## ##### ## . . . # ## . . . # #####	6
10 8 #### . . ## # . . # . ## . # . . # . ### #### . # . # ... ##### .###.### ###.### ##### ###.### .###.###.	18

Замечание