

В. Сбалансированные таблицы

Входной файл:	<i>b.in</i>
Выходной файл:	<i>b.out</i>
Время на тест:	1 секунда

Пусть $A[1..N, 1..N]$ - таблица, элементами которой являются целые числа. Рассмотрим $p[1..N]$ - произвольную перестановку натуральных чисел от 1 до N . Определим величину $S(p) = A[1, p[1]] + A[2, p[2]] + \dots + A[N, p[N]]$. Таблицу A будем называть сбалансированной, если значение $S(p)$ не зависит от выбора p .

Напишите программу, которая определяет, является ли заданная таблица A сбалансированной.

Входные данные. Сначала во входном файле записано натуральное число N ($1 \leq N \leq 20$) - размер таблицы A . Далее следуют элементы таблицы A , которые перечисляются по строкам сверху вниз. Элементы каждой строки перечисляются слева направо. Каждый элемент таблицы по модулю не превосходит 100. Числа во входном файле разделяются пробелами и/или переводами строки.

Выходные данные. Выведите в выходной файл единственную строку. Если таблица A является сбалансированной, то эта строка должна содержать слово "YES", в противном случае - слово "NO".

Примеры.

<i>Input file</i>	<i>Output file</i>
2 0 0 0 0	YES
2 0 0 0 1	NO

С. Третья проблема Полигонии

Входной файл:	<i>c.in</i>
Выходной файл:	<i>c.out</i>
Время на тест:	1 секунда

Страна Полигония имеет форму правильного N -угольника. В каждой вершине этого многоугольника находится дом ровно одного жителя страны. Два дома называются соседними, если отрезок, соединяющий их, лежит целиком на границе страны. Между каждой парой домов, не являющихся соседними, есть улица - отрезок, соединяющий эти два дома. Точки плоскости, которые принадлежат более чем одной улице и не являются домами, называются площадями.

Однажды король страны поставил перед согражданами сложную задачу, известную теперь, как первая проблема Полигонии. Она звучала следующим образом: "Чему равно количество домов в Полигонии?" Спустя столетия эта задача была решена. Оказалось, что в Полигонии ровно N домов.

После этого король (уже другой) придумал еще более сложную задачу: "Чему равно количество улиц в Полигонии?". Эта задача была окрещена жителями второй проблемой Полигонии. На ее решение понадобились тысячелетия, но и эта задача была решена. Оказалось, что улиц в Полигонии ровно в два раза меньше, чем $(N-1)(N-2)-2$.

Наконец, очередной король придумал, как ему показалось, неразрешимую задачу: “Чему равно количество площадей в Полигонии?”. Как Вы уже догадались, эта задача - третья проблема Полигонии. Несколько миллионов лет лучшие умы страны бились над этой задачей и не могли ее решить. Это породило множество упрощенных вариантов третьей проблемы Полигонии. Один из них звучал следующим образом: “Пусть в Полигонии S площадей. Чему равен остаток от деления S на N ?”.

Вам предстоит сделать невозможное и решить третью проблему Полигонии в описанном выше упрощенном виде.

Входные данные. Единственная строка входного файла содержит целое число N ($3 \leq N \leq 100$) - количество домов в Полигонии.

Выходные данные. Единственная строка выходного файла должна содержать одно целое число - остаток от деления числа S на N , где S - количество площадей в Полигонии.

Примеры.

<i>Input file</i>	<i>Output file</i>
3	0
4	1

D. Мафия бессмертна

<i>Входной файл:</i>	<i>d.in</i>
<i>Выходной файл:</i>	<i>d.out</i>
<i>Время на тест:</i>	<i>1 секунда</i>

В городе Усть-Урюпинске живет N жителей и действует K различных валют. Каждый житель имеет некоторые сбережения. Известно, что сбережения i -го жителя в j -й валюте равны A_{ij} валютных единиц.

Все было хорошо в Усть-Урюпинске, и все жители жили долго и счастливо. Но однажды в городе появилась мафия!!! Да не одна мафиозная группировка, а сразу несколько.

Обеспокоились жители Усть-Урюпинска. Не знают, кому верить, а кому – нет. Не знают, кто из их знакомых – честный и порядочный гражданин, а кто – злобный мафиози. Но, по крайней мере, один честный человек остался. Это – Антонина Гавриловна.

Антонина Гавриловна долго изучала законы, по которым живет мафия, и теперь она точно знает следующее:

- 1) В каждой мафиозной группировке состоят ровно 2^t человек.
- 2) Суммарные сбережения каждой из группировок в каждой из валют делятся на 2^t . Под суммарными сбережениями группировки в некоторой валюте подразумевается сумма всех сбережений людей, состоящих в группировке, в этой валюте.
- 3) Количество мафиозных группировок не менее, чем $\lfloor N/2^t \rfloor - 2^K$.

Понятно, что по этим данным нельзя однозначно восстановить состав каждой из группировок. Поэтому Антонина Гавриловна хочет найти хотя бы какое-нибудь распределение жителей по мафиозным группировкам, удовлетворяющее этим трем условиям.

Напишите программу, которая поможет Антонине Гавриловне. Вам следует учесть, что Антонина Гавриловна могла допустить ошибку в изучении законов мафии, и такого распределения может не существовать. Учтите также, что некоторые жители могут не принадлежать никакой мафиозной группировке. Наконец, примите к сведению, что сама Антонина Гавриловна не является жителем Усть-Урюпинска.

Входные данные. Первая строка входного файла содержит числа N , K и t ($1 \leq N \leq 10000$; $1 \leq K, t \leq 10$), разделенные пробелами. Каждая из следующих N строк содержит по K целых чисел разделенных пробелами. j -е число в i -й строке равно A_{ij} ($1 \leq A_{ij} \leq 10000$).

Выходные данные. Если распределения жителей по группировкам, удовлетворяющего всем приведенным условиям, не существует, то выведите в первой строке выходного файла слово «NO». В противном случае выведите в первой строке слово «YES». Если распределение жителей по группировкам существует, то следующие N строк должны описывать найденное Вами распределение (если их несколько, то любое). i-я из этих строк должна содержать номер группировки, в которой состоит i-й житель, или число 0, если этот житель не состоит ни в какой из группировок. Нумеруйте группировки последовательными натуральными числами, начиная с 1.

Примеры.

<i>Input file</i>	<i>Output file</i>
4 3 1 1 2 3 4 5 6 7 8 9 10 11 12	YES 1 2 1 2
4 3 1 1 2 3 4 5 6 7 8 9 10 11 12	YES 0 0 0 0

Е. Квадроквадратуры

<i>Входной файл:</i>	<i>e.in</i>
<i>Выходной файл:</i>	<i>e.out</i>
<i>Время на тест:</i>	<i>1 секунда</i>

Упорядоченную четверку натуральных чисел (a, b, c, d) будем называть квадроквадратурой натурального числа N, если выполняется следующее равенство:

$$a^2 + b^2 + c^2 + d^2 = N$$

Напишите программу, которая находит количество квадроквадратур заданного натурального числа N.

Входные данные. Входной файл содержит одно натуральное число N ($1 \leq N \leq 1000000$).

Выходные данные. Выведите в выходной файл одно число - количество квадроквадратур числа N.

Примеры.

<i>Input file</i>	<i>Output file</i>
3	0
4	1

Г. Побег из тюрьмы

<i>Входной файл:</i>	<i>f.in</i>
<i>Выходной файл:</i>	<i>f.out</i>
<i>Время на тест:</i>	<i>1 секунда</i>

Недавно вышедшая компьютерная игра «Побег из тюрьмы» просто захватывает дух! Вы управляете заключенным, который должен убежать из тюрьмы. Территория, по которой двигается заключенный, охраняется солдатами и вертолетами, которые постоянно двигаются и в любой момент могут Вас заметить. Петя Булочкин – бывалый игрок, который за свою жизнь прошел много компьютерных игр. Однако он утверждает, что, проектируя первый

уровень, разработчики «Побега из тюрьмы» допустили ошибку, и этот уровень пройти невозможно. Напишите программу, которая проверяет, прав ли Петя Булочкин. Рассмотрим математическую модель первого уровня игры «Побег из тюрьмы». Территория тюрьмы представляет собой бесконечную во всех направлениях плоскость. На территории есть два заграждения из колючей проволоки. Первое заграждение пролегает вдоль прямой $y = 0$, а второе – вдоль прямой $y = L$. Заключенный не может переходить через эти заграждения. В начальный момент времени заключенный находится в точке с координатой $(-12345.6789, L/2)$. Его цель – попасть в точку с координатой $(98765.4321, L/2)$. Заключенного в тюрьме очень плохо кормят, поэтому он очень худой и в нашей модели мы будем считать его сколь угодно малой точкой.

Выполнить цель заключенному мешает солдат, который ходит по отрезку, соединяющему точки $(0, 0)$ и $(0, L)$. Дойдя до одного из концов отрезка, солдат сразу же разворачивается и идет в противоположную сторону. Солдат движется равномерно со скоростью U . Если заключенный в некоторый момент времени оказывается в точке, удаленной от точки, в которой в этот момент времени находится солдат, менее, чем на R , то солдат видит заключенного, что означает поражение игрока. Заключенный может двигаться с произвольной, не обязательно все время одинаковой скоростью, не превосходящей V .

Входные данные. Первая и единственная строка входного файла содержит четыре целых числа L, U, R, V ($1 \leq L, U, R, V \leq 1000$) в указанном порядке, разделенные одним или несколькими пробелами.

Выходные данные. Выведите в выходной файл слово «YES», если предположение Пети Булочкина верно, то есть пройти первый уровень игры «Побега из тюрьмы» невозможно. Выведите в выходной файл слово «NO» в противном случае.

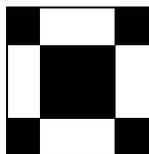
Примеры.

<i>Input file</i>	<i>Output file</i>
100 1000 1000 1	YES
100 1 1 1000	NO

Г. Эквивалентные изображения

<i>Входной файл:</i>	<i>g.in</i>
<i>Выходной файл:</i>	<i>g.out</i>
<i>Время на тест:</i>	<i>1 секунда</i>

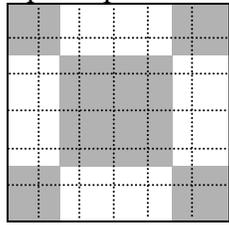
В этой задаче мы будем работать с прямоугольными изображениями, каждый пиксел которых может иметь один из двух цветов – черный или белый. Примером такого изображения является следующее:



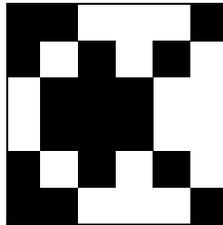
Рассмотрим следующие три операции над изображениями:

- 1) *Симметрия.* Применение этой операции заключается в симметричном отображении изображения относительно горизонтальной или вертикальной оси.
- 2) *Поворот.* Применение этой операции заключается в повороте изображения на 90° по часовой стрелке.
- 3) *Масштабирование.* Применение этой операции происходит в несколько этапов. Рассмотрим его на примере изображения на рисунке сверху. Это изображение имеет размеры 4×4 . Вначале выбираются произвольные новые размеры изображения,

например, $6 * 6$. Затем поверх изображения рисуется равномерная сетка, имеющая выбранные размеры.



Новое изображение формируется по следующему принципу: для каждой клетки нарисованной сетки смотрится, какого в ней цвета больше – черного или белого. Если черного цвета больше, то соответствующий пиксел нового изображения красится в черный цвет. Если белого цвета больше, то соответствующий пиксел нового изображения красится в белый цвет. Наконец, если площади покрытия клетки обоими цветами совпадают, то соответствующий пиксел нового изображения может быть окрашен как в белый, так и в черный цвет. После применения операции масштабирования новое изображение может выглядеть, к примеру, следующим образом:



Заметим, что действие операции масштабирования не является однозначным. Даже при одинаковом выборе новых размеров изображения с помощью масштабирования можно получать различные изображения.

Два изображения называются эквивалентными, если одно из них можно получить из другого с помощью последовательности операций описанного вида, среди которых не более одной операции масштабирования. Напишите программу, которая проверяет, являются ли два заданных изображения эквивалентными.

Входные данные. Первая строка входного файла содержит два натуральных числа N и M ($1 \leq N, M \leq 100$) – размеры первого изображения. Следующие N строк описывают изображение. Каждая из этих строк содержит ровно M символов «0» и/или «1». j -й символ в i -й строке кодирует цвет j -го пикселя в i -й строке изображения. Символ «0» кодирует белый цвет, а символ «1» – черный. Далее в файле в точно таком же формате описывается второе изображение.

Выходные данные. Единственная строка выходного файла должна содержать слово «YES», если изображения во входном файле эквивалентны, и слово «NO» в противном случае.

Примеры.

<i>Input file</i>	<i>Output file</i>
<pre>4 4 1001 0110 0110 1001</pre>	YES
<pre>6 6 110001 101010 011100 011100 101010 110001</pre>	

1 1	NO
0	
1 1	
1	

Н. Лабиринт

<i>Входной файл:</i>	<i>h.in</i>
<i>Выходной файл:</i>	<i>h.out</i>
<i>Время на тест:</i>	<i>1 секунда</i>

Рассмотрим следующую игру "Лабиринт". Для этой игры необходимо прямоугольное клеточное поле с N строками и M столбцами. Две клетки поля будем называть соседними, если они имеют общую сторону. Некоторые клетки поля закрашены в черный цвет, остальные клетки - белые. Исключение составляют две клетки, одна из которых красная, а вторая - синяя. Дополнительно, для игры нужны фишки красного, синего и белого цветов.

"Лабиринт" - игра для двух игроков. В начале игры ходит первый игрок. Он ставит белую фишку на красную клетку и начинает делать ходы. За один ход он должен передвинуть фишку в одну из 4-х соседних клеток. Кроме того, клетка, в которую передвигается фишка, не может иметь черный цвет. Первый игрок заканчивает делать свои ходы, как только белая фишка попадает на синюю клетку. После этого белая фишка убирается с поля.

Далее ходит второй игрок. Он ставит красную фишку на красную клетку и синюю фишку на синюю клетку. Затем второй игрок начинает свою серию ходов. За один ход он должен передвинуть каждую из фишек в любую из 4-х соседних клеток. Как и первый игрок, второй игрок не может двигать никакую из своих фишек на клетку черного цвета. Кроме этого, есть еще два ограничения. Запрещается делать такие ходы, после которых красная и синяя фишки оказываются в одной и той же клетке. Наконец, если красная и синяя фишки в некоторый момент игры стоят на соседних клетках, то запрещается делать ход, приводящий к обмену фишек местами. Второй игрок заканчивает делать свои ходы, как только после очередного хода красная фишка оказывается на синей клетке, а синяя фишка - на красной клетке.

Если первый игрок не видит способа завершить свою серию ходов, то он может сдаться. В этом случае победа присуждается второму игроку. Аналогично, если второй игрок не видит способа завершить свою серию ходов, то он также может сдаться. В этом случае победа присуждается первому игроку. В случае, когда каждый из игроков завершил свои ходы и не сдался, победитель определяется следующим образом. Пусть $C1$ - количество ходов, сделанных первым игроком, а $C2$ - количество ходов, сделанных вторым игроком. Если $C2 \leq C1$, то победителем считается второй игрок, в противном случае - первый игрок.

По заданному описанию поля для игры в "Лабиринт" определите, кто будет победителем в игре. Считайте, что оба игрока играют оптимально.

Входные данные. Первая строка входного файла содержит два целых числа N и M ($1 \leq N$, $M \leq 200$), разделенные одним пробелом. Следующие N строк описывают игровое поле. Каждая из этих строк содержит ровно M символов. j -й символ в i -й строке обозначает цвет клетки поля на пересечении i -й строки и j -го столбца. Он может быть равен 'B', 'W', 'R', 'S'. Символ 'B' означает черный цвет, символ 'W' - белый цвет, символ 'R' - красный цвет, символ 'S' - синий цвет.

Выходные данные. Выведите в выходной файл одно число, равное 1, если победителем будет первый игрок, и равное 2 в противном случае.

Примеры.

<i>Input file</i>	<i>Output file</i>
3 3 WRW WBW WSW	2
3 3 WWW WBW RWS	1