

Задача А: Реформы Северуса Снейпа

Входной файл: **psycho.in**
Выходной файл: **psycho.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

На выборах нового ректора Хогвартса победил профессор Северус Снейп. Его программа действий заключалась в отходе от устаревших традиций и приведении системы образования в Хогвартсе в соответствие с современными идеями (как магическими, так и маггловскими).

Чтобы переход был постепенным, Снейп принял решение о формировании в каждом колледже экспериментальных групп, которые будут работать по интенсивной методике, изучая, помимо традиционных для Хогвартса дисциплин, ещё и программирование. Как сказал Снейп, «мы, маги, должны быть выше магглов во всём, а сейчас Хогвартс отсутствует даже в расширенном списке призёров IOI»

Из N кандидатов, рекомендованных в экспериментальные группы, выбирается состав учеников так, чтобы туда не попало ни одной пары людей, психологически несовместимых друг с другом – это приведет к постоянным конфликтам и снижению эффективности обучения.

Предварительное тестирование выявило K пар кандидатов, несовместимых друг с другом. При этом был отмечен удивительный факт: среди любой четверки кандидатов либо нет ни одной пары несовместимых людей, либо выявлено больше, чем одна такая пара.

Вы приглашены Снейпом вести курс программирования в этих группах. Для начала перед Вами поставлена задача: подобрать в состав группы как можно большее количество кандидатов, среди которых нет ни одной пары несовместимых между собой людей.

Входные данные: Первая строка входного файла содержит величины N ($4 \leq N \leq 1000$) и K ($0 \leq K \leq N * (N-1) / 2$). Каждая из K последующих строк содержит два номера, разделенные пробелами – номера психологически несовместимых учащихся (нумерация начинается с единицы, второй номер в паре больше первого).

Выходные данные содержат единственное число – искомое количество испытателей.

<u>psycho.in</u>	<u>psycho.out</u>
6 8	3
2 4	
1 6	
2 5	
1 4	
1 5	
3 6	
2 6	
3 5	

Пояснение: В экспериментальную группу можно рекомендовать кандидатов с номерами 1, 2, 3; либо 4, 5, 6.

Задача В. Трансфигурация

Входной файл: **cheval.in**
Выходной файл: **cheval.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

Темой лекции профессора МакГонагалл по трансфигурации было непрерывное превращение. В качестве исходного материала она использовала магические шахматы.

Назовем шахматную фигуру (n,k) -конем, если она перемещается на n клеток по горизонтали или вертикали и на k клеток в перпендикулярном направлении. Обычный шахматный конь является, согласно этой терминологии, $(2,1)$ -конем.

В начале эксперимента на шахматную доску помещена единственная фигура – $(1,0)$ -конь. Первоначально она находится на клетке **A**. Результат применения заклинания непрерывного превращения заключается в том, что после первого хода эта фигура превращается в $(1,1)$ -коня, затем – в коня с характеристиками $(2,1)$, $(2,2)$, $(3,2)$ и т.д.

Рон, попавший в экспериментальную группу, изучавшую программирование и откровенно скучавший на лекциях, не относившихся к квиддичу и программированию, задумался, а за какое минимальное количество ходов такая фигура достигнет клетки **B**, не совпадающей с **A**. Выходить за границы доски нельзя!

Напишите программу, отвечающую на вопрос Рона.

Входной файл. Единственная строка этого файла содержит координаты клеток **A** и **B** в стандартной шахматной нотации, разделенные одним пробелом. Согласно этой нотации, столбцы доски обозначаются латинскими буквами от a до h слева направо, строки – цифрами от 1 до 8 снизу вверх, так что левая нижняя клетка обозначается как a1, а правая верхняя – как h8.

Выходные данные содержат единственное число – минимальное количество ходов, необходимое для перемещения фигуры из клетки A в клетку B, либо -1, если решения задачи не существует.

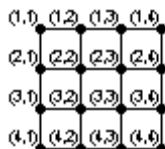
cheval.in c4 d5	cheval.out 3
---------------------------	------------------------

Пояснения: один из возможных путей – c4, d4, c3, d5.

Задача С. Подземелья Хогвартса.

Входной файл: **coloring.in**
Выходной файл: **coloring.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

План подземелий Хогвартса можно представить как граф G_N , состоящий из N^2 вершин, каждая из которых соответствует некоторой паре (i, j) , $1 \leq i, j \leq N$, и $2N(N-1)$ ребер, соединяющих вершины (i, j) и (i', j') , для которых выполнено равенство $|i - i'| + |j - j'| = 1$. Например, для $N = 4$:



Каждая пещера, таким образом, является вершиной графа, а каждый коридор – его ребром. Новый директор собирается осветить часть коридоров таким образом, чтобы существовал такой способ включить свет в каждой из комнат, что освещены только те коридоры, которые соединяют две освещённые комнаты (при этом освещение в комнатах и коридорах независимо, освещение (или его отсутствие) для каждого из коридоров постоянно).

Напишите программу, получающую на вход план частично освещённых подземелий т.е. план, в котором освещённость части коридоров известна, а части – нет, и возвращающую количество способов освещения неизвестных коридоров, удовлетворяющую требованиям Снейпа.

Входные данные.

Первая строка входного файла содержит целое число N ($1 < N \leq 10$).

В следующих $2N-1$ строках описывается частичная раскраска ребер графа G_N (формат описания – в примере ввода). Различные символы в описании имеют следующие значения:

- ‘.’ – комната;
- ‘R’ –освещённый коридор;
- ‘G’ – неосвещённый коридор;
- ‘?’ – коридор, по поводу которого решение не принято.

Выходные данные.

Целое число, равное искомому количеству способов.

Пример.

coloring.in	coloring.out
4	158
.??.?.	
R ? ? G	
.??.?.	
? R G ?	
.??.?.	
R ? ? G	
.??.?.	

Задача D. Помощь угнетённым колдовским народам

Входной файл: **dish.in**
Выходной файл: **dish.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

Пытаясь донести до сознания соучеников по Хогвартсу всю тяжесть положения домовых эльфов, Гермиона Грейнджер решила сделать плакат, показывающий, насколько тяжела их работа. Один из примеров - после обеда в столовой Хогвартса N грязных тарелок составлены в стопку доньями вниз и ожидают мойки... Гермиона планирует снабдить плакат точными данными о соотношении высоты стопки тарелок и среднего роста домовых эльфов. Но для начала надо подсчитать эту высоту – но проблема в том, что Гермиона очень сильно занята и у неё нет времени написать программу. Можете ли Вы помочь угнетённым народам и подсчитать высоту этой стопки?

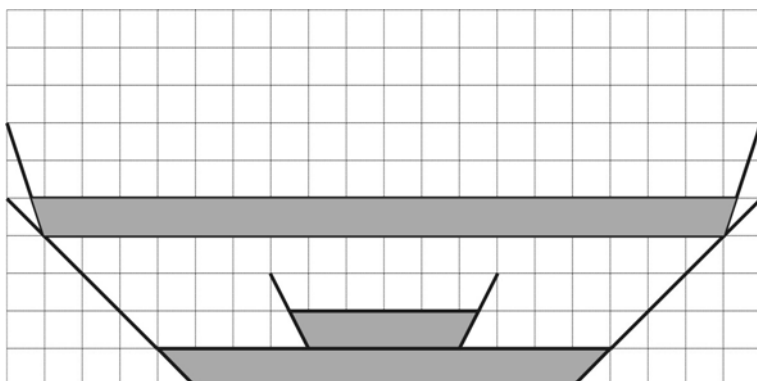
Каждая тарелка представляет собой усеченный конус, причем дно тарелки находится в основании с меньшим радиусом и имеет толщину 1, а толщиной боковых стенок можно пренебречь. Тарелки составлены так, что существует вертикальная прямая, проходящая через центры оснований всех конусов.

Входные данные. Первая строка содержит значение N ($2 \leq N \leq 10000$). Каждая из последующих N строк описывает одну тарелку и содержит три целых числа – радиус дна R_1 , радиус верхней части R_2 и высоту тарелки H ($1 \leq R_1 < R_2 \leq 1000000$, $2 \leq H \leq 1000000$). Числа в строках входного файла разделены одним или несколькими пробелами. Описание тарелок соответствует их расположению в стопке снизу вверх.

Выходные данные. Единственная строка выходного файла должна содержать искомую высоту стопки, рассчитанную с точностью до 0.001.

dish.in	dish.out
3	7.000
5 10 5	
2 3 2	
9 10 3	

Пояснение: соответствующее примеру расположение тарелок изображено на рисунке



Задача E. Вандализм

Входной файл: **ink.in**
Выходной файл: **ink.out**
Time Limit: **8 секунд**
Memory Limit: **64 М**

Драко Малфой, не прошедший из-за своей конфликтности в экспериментальную группу, решил устроить очередную пакость. Он одновременно нанёс на одну из картин в гриффиндорской башне N капель чернил. После нанесения на бумагу чернильное пятно начинает растекаться по листу равномерно во всех направлениях до тех пор, пока не соприкоснется с краем картины или с другим пятном. Скорость растекания чернил одинакова для всех капель.

Однако юный вандал был пойман на месте преступления и препровождён к директору. Ссылки на свободу самовыражения и творчество художника Калевича не помогли: Снейп принял решение: снять с Слизера 239 баллов, а стоимость реставрации картины взыскать с Малфоев. Для определения этой стоимости требуется определить площадь повреждённого участка после того, как растекание всех капель прекратится.

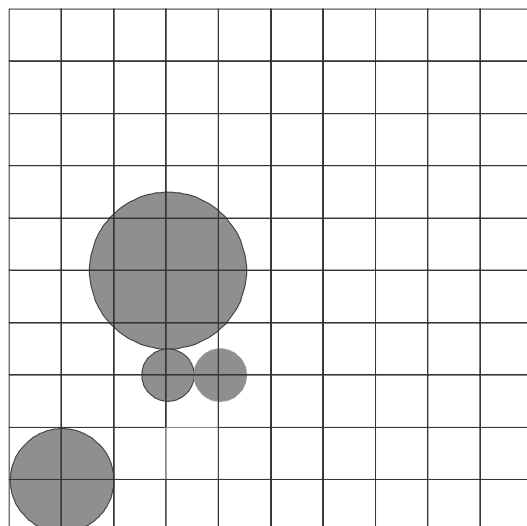
Входные данные: Первая строка входного файла содержит значения N (количество капель), W и H (соответственно ширина и высота листа). Каждая из последующих N строк описывает координаты x и y точки, в которую изначально помещается капля чернил. Все эти точки не совпадают друг с другом, и ни одна из них не находится на краю листа. Все числа целые и отделяются друг от друга одним или несколькими пробелами.

Ограничения на входные данные: $1 \leq N \leq 500$; $2 \leq W, H \leq 10000$, $0 < x < W$, $0 < y < H$.

Выходные данные содержат единственное число – искомую площадь повреждённого участка, рассчитанную с точностью до 0.001.

<u>ink.in</u>	<u>ink.out</u>
4 10 10 1 1 3 3 4 3 3 5	11.781

Пояснение: соответствующее примеру изображение показано на рисунке.



Задача F. Обмен колдовским опытом

Входной файл: **natural.in**
Выходной файл: **natural.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

В порядке изучения колдовских традиций различных народов делегация Хогвартса во главе с профессором Флитвиком устроила экспедицию в некоторую отдалённую страну. Но на пути экспедиции возникла проблема: если жители городов еще умеют пользоваться деньгами, то племена, контролирующие торговые пути между городами, не слышали, что такое сикли или галлеоны и предпочитают оплату натурой. Иными словами, владелец каравана, следующего из одного города в другой, должен отдать в качестве платы за проход определенное количество шкур, руды, огненной воды или других предметов, нужных племени. К счастью, эти предметы можно купить за золото в городах, через которые проходит караван.

Вам необходимо провести караван экспедиции из одного города в другой, затратив на плату за проход между городами минимальное количество золота. Сможете ли Вы сделать это?

Входные данные: Первая строка содержит параметры задачи: значения **N** (количество городов в стране, пронумерованных, начиная с единицы), **M** (количество торговых путей) и **K** (количество предметов, используемых в качестве платы за проезд) Ограничения на параметры задачи: $2 \leq N \leq 1000$, $1 \leq M \leq 10000$, $1 \leq K \leq 10$. Каждая из последующих **N** строк входного файла описывает цену предметов, используемых в качестве платы за проход, в этом городе. Строка содержит **K** неотрицательных чисел, не превосходящих 1000 – цену единицы каждого предмета. Если какое-либо число равно 0, соответствующий предмет нельзя купить в этом городе, в противном случае количество предметов, которые можно купить, не ограничивается.

Наконец, каждая из последующих **M** строк файла описывает один торговый путь и содержит **K+2** неотрицательных числа: номер города – пункта отправления, номер города – пункта назначения и количество предметов каждого вида (не превосходящее 1000), которые необходимо отдать в качестве платы за проход по этому пути. Если количество какого-либо предмета в описании пути равно 0, соответствующий предмет нельзя использовать в качестве оплаты прохода по этому пути. Если эта величина равна 0 для всех возможных предметов, плата за проход по данному пути не требуется. Все пути – односторонние, но из одного города в другой может существовать несколько различных путей.

Вам необходимо провести караван экспедиции из города 1 в город **N**. Выбрав один из путей, соединяющих два города, Вы не можете в дальнейшем свернуть с него до прихода в пункт назначения. При оплате за проход по любому пути Вы можете использовать только один из предметов, необходимых племени (на Ваш выбор). В начале пути никаких предметов, используемых в качестве оплаты, у экспедиции нет, но Вы можете сделать первую закупку в городе 1. Все числа – целые и отделяются друг от друга одним или несколькими пробелами.

Выходные данные: Единственная строка выходного файла должна содержать минимальное количество золота, которое необходимо истратить в качестве платы за проход. Если задача не имеет решения, вы должны вывести в выходной файл значение -1.

<u>Пример входных данных</u>	<u>Пример выходных данных</u>
3 4 2 1 2 2 1 0 1 1 2 2 0 2 1 2 0 2 3 15 10 1 3 10 5	9

Пояснение: Обозначим предметы оплаты через А и В. Вы должны купить в городе 1 четыре предмета А, перейти в город 2, купить там пять предметов В и вернуться в город 1. Теперь можно перейти в город 3, расплатившись за этот проход предметами В.

Задача G. Перед четвертьфиналами.

Входной файл: **train.in**
Выходной файл: **train.out**
Time Limit: **2 секунды**
Memory Limit: **64 М**

Реформы Снейпа коснулись и квиддича. Первенство Хогвартса было расширено до 8 команд (по 2 от каждого колледжа), а вместо мадам Хуч был выписан голландский тренер по квиддичу, владевший передовыми методиками «тотального квиддича». Для того, чтобы игроки могли набирать оптимальную форму, необходим тренажерный зал. Но методы, использующие магию, с помощью сочетания физической нагрузки и приёмов непрерывной трансфигурации позволяют игрокам за сколь угодно малое время набрать или сбросить произвольное количество килограммов веса. Для того, чтобы посетить тренажерный зал, игрок должен заранее подать заявку, в которой указаны:

- время прихода клиента T_1
- время ухода клиента T_2 ($T_2 > T_1$)
- текущий вес клиента W_1
- желаемый вес клиента W_2

Во время работы в тренажерном зале нагрузка игрока будет определена таким образом, чтобы его вес за время от T_1 до T_2 равномерно изменялся от W_1 до W_2 , таким образом в момент времени t , $T_1 \leq t \leq T_2$, вес игрока будет равен $W_1 + (t - T_1) * (W_2 - W_1) / (T_2 - T_1)$.

Проблема в том, что игроки в квиддич – очень импульсивные люди. В силу своей импульсивности они очень любят отрабатывать друг на друге силовые единоборства (в простонародье называемые драками), и это стало настоящей проблемой для тренировочного процесса. Чтобы решить проблему, голландец предложил Снейпу нанять специального охранника.

Оплата охранника – почасовая, поэтому Снейп решил использовать его услуги как можно эффективнее. Понаблюдав за игроками, он сделал вывод, что два игрока различной массы никогда не вступают друг с другом в силовое единоборство (в силу очевидности победы одного из них второй старается держаться от первого подальше). Поэтому в услугах охранника школа нуждается только в такие моменты времени t , когда как минимум два из находящихся в тренажерном зале игроков имеют одинаковую массу. Теперь вечером каждого дня по имеющему списку заявок на следующий день определяется $\min T$ – минимальный момент времени, когда два или более игрока в тренажерном зале имеют одинаковый вес, после чего администрация сообщает охраннику, что завтра ему надо придти на работу в момент времени $\min T$.

Так как близится начало четвертьфиналов очередного квиддичного сезона, то услуги тренажерного зала становятся все более и более востребованными, соответственно, игроков становится все больше и больше. Теперь определять момент времени $\min T$ из-за возросшего количества заявок стало очень сложно, и Вам поручили написать для его определения программу.

Входные данные

Первая строка входного файла содержит целое число N ($1 \leq N \leq 55555$) – количество заявок на следующий день.

Следующие N строк описывают заявки. Каждая из заявок описывается четырьмя целыми числами T_1 , T_2 , W_1 , W_2 ($1 \leq T_1 < T_2 \leq 100000$; $1 \leq W_1, W_2 \leq 100000$), смысл которых описан в условии задачи.

Выходные данные

Единственная строка выходного файла должна содержать минимальный из моментов времени, когда два или более игроков в тренажерном зале будут иметь одинаковый вес (гарантируется, что хотя бы один из таких моментов существует). Абсолютная погрешность ответа не должна превышать 10^{-3} .

Примеры

train.in	train.out
2 1 3 1 3 3 5 3 1	3.0000000000
2 90000 95000 900 950 92500 100000 925 1000	92500.0000000000
4 2 11 2 7 2 15 10 4 8 10 10 2 1 10 5 5	7.4000000000

Пояснения: возможные расстановки оставшегося 4-палубного корабля показаны на рисунках (этот корабль обозначен символами 'X')

<pre>*.*.*. ...*.*.* .**...*. ...*...* ...*..X... ...*.X.*... ...*.X.*... ...*.X.*...X..... </pre>	<pre>*.*.*. ...*.*.* .**...*. ...*... X.....* X.*..... X.*...*... X.*...*... </pre>	<pre>*.*.*. ...*.*.* .**...*. ...*...* X.*..... X.*...*... X.*...*... X.*...*... X..... </pre>
--	---	--

Задача I. Конкурс Itransition в Хогвартсе

Входной файл: **suffix.in**
Выходной файл: **suffix.out**
Time Limit: **3 секунды**
Memory Limit: **64 М**

Эксперимент Снейпа по обучению выпускников Хогвартса программированию был признан удачным. Магические приёмы успешно применялись для отладки программ, из-за чего маги-программисты имели преимущество. И вот активно сотрудничающая с Министерством Магии компания **Itransition** открыла одну вакансию мага-программиста. Испытанием для соискателей была задача построения суффиксного дерева.

В этой задаче необходимо по заданному корневому дереву T определить количество различных строк, состоящих из символов '0'/'1', суффиксные деревья которых изоморфны T .

Две строки называются различными, если они имеют различную длину, или, если существует такое i , что i -й символ первой строки не равняется i -му символу второй строки.

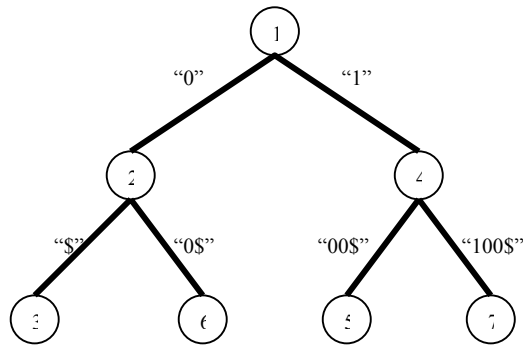
Два корневых дерева называются изоморфными, если они состоят из одинакового количества вершин N и можно пронумеровать вершины каждого из них различными натуральными числами от 1 до N так, что:

- 1) корни обоих деревьев имеют одинаковые номера;
- 2) первое дерево содержит ребро между вершинами с номером a и b тогда и только тогда, когда второе дерево также содержит ребро между вершинами с номерами a и b .

Суффикс строки S – это подстрока, начинающаяся в ее произвольном символе и заканчивающаяся в ее последнем символе. Суффиксное дерево строки S – это корневое дерево, удовлетворяющее следующим правилам:

1. Каждому ребру дерева поставлена в соответствие непустая строка, состоящая из символов "0"/"1".
2. Каждому суффиксу строки $S+'$'$, кроме суффикса "\$", соответствует ровно один лист (т.е. вершина степени 1, не совпадающая с корнем) такой, что конкатенация строк ребер на пути от корня к этому листу равна данному суффиксу.
3. Степень любой вершины, не являющейся ни листом, ни корнем дерева, строго больше двух.
4. Дерево имеет минимальную возможную суммарную длину строк, записанных на ребрах, среди всех деревьев, удовлетворяющих свойствам 1-3.

Например, суффиксное дерево для строки "1100" выглядит следующим образом:



Компании Itransition для успешного проведения экзамена потребовалось эталонное решение, написать которое поручено Вам.

Входные данные

Первая строка входного файла содержит целое число N – количество вершин дерева ($2 \leq N \leq 25$). Вершины пронумерованы натуральными числами от 1 до N . Корнем дерева является вершина с номером 1.

В следующих $N-1$ строках описываются ребра дерева. Каждая из этих строк содержит два целых числа X и Y – номера вершин, которые соединяет очередное ребро дерева.

Выходные данные

Первая строка выходного файла должна содержать целое число C – количество строк, суффиксное дерево которых изоморфно дереву, описанному во входном файле (гарантируется, что хотя бы одна такая строка существует).

В следующих C строках файла должны быть выведены все искомые строки. Порядок их вывода может быть произвольным.

Пример

suffix.in	suffix.out
7	6
1 2	0011
1 4	0101
2 6	1001
2 3	0110
4 5	1010
4 7	1100