

Задача А

Путь путника

В этой задаче всё просто, как в школьных задачах по физике. Путник вышел из пункта А и двигается в направлении пункта В с постоянной скоростью V . Одновременно с выходом путника могильщик начал копать могилу в пункте В, углубляясь в землю со скоростью V_h . Как только путник дойдёт до пункта В, могильщик перестанет копать и мгновенно выскочит из могилы, дабы не мешать свободному падению путника в свежеврытую могилу.

Входные данные (way.in):

В первой строке дано целое число – расстояние S между пунктами А и В в метрах ($1 \leq S \leq 1000$). Во второй строке через пробел дано ещё два целых числа: скорость путника и скорость углубления могильщика в метрах в секунду, оба числа целые ($1 \leq V, V_h \leq 1000$).

Выходные данные (way.out):

Выведите одно число – путь путника в метрах. Путь выведите точно с 6 знаками после запятой.

Пример:

way.in	way.out
100 3 1	133.333333

Задача В

Сумматор

- Сколько будет, если сложить два числа?
- А числа большие?
- Очень!

Входные данные (summator.in):

Во входном файле находится два целых неотрицательных десятичных числа без ведущих нулей. Числа записаны через пробел и каждое из них не превосходит величины 10^{100} .

Выходные данные (summator.out):

Выведите сумму этих чисел в том же формате, как и слагаемые во входном файле.

Пример:

summator.in	summator.out
10001 9999	20000
1 0	1
0 0	0

Задача С

Земля

В советские времена народу выделялось несколько соток земли, или другими словами, давались дачи. Вдоль одной из просёлочных дорог, как ни странно, прямой, начальник дач наставил колышки через одинаковое расстояние, дабы поделить всё поровну.

Позже «сверху» пришло указание, сделать участки треугольной формы. И местное руководство вбило ещё один колышек на некотором расстоянии от дороги и утвердило, что это и будет третья вершина для всех участков. А Вас направили определить

«счастливчика» – первого в списке, у которого участок по площади не превысит площадь любого другого участка из этого же списка.

Входные данные (ground.in):

В первой строке единственное целое число N ($2 \leq N \leq 1000$) – количество участков. В следующей строке дано два числа с точностью не более 3 знаков после десятичной точки – координаты общей вершины (x, y) .

В последующих N строках дано по четыре числа (x_2, y_2, x_3, y_3) – вторая и третья вершина каждого из участков.

Все числа во входном файле даны с точностью не более 3 знаков после десятичной точки и по модулю не превосходят 10000000.

Выходные данные (ground.out):

Вывести порядковый номер «счастливчика».

Пример:

ground.in	ground.out
2	1
0 10	
-10 0 0 0	
10 0 0 0	

Задача D

Строки

Для некоторых алгоритмов сжатия текстовых строк необходимо знать частоту, с которой встречается та или иная буква в тексте. Вот для Вас и поставлена задача: определить, сколько раз встречается в тексте та или иная буква.

Входные данные (string.in):

Дана одна строка длиной не более 50000 букв (только латинские строчные буквы).

Выходные данные (string.out):

Вывести буквы в порядке увеличения частоты встречи их в тексте, а при одинаковой частоте – в порядке их появления в строке (см. примеры).

Пример:

string.in	string.out
aaacbbzc	zcbbaaa
abcabbca	ccababba
ababcdtcddtaabbcd	ttcdcdcdababaabb

Задача E

Ноев ковчег

«И увидел Господь, что велико развращение человеков на земле, и что все мысли и помышления сердца их были зло во всякое время; и раскаялся Господь, что создал человека на земле, и восскорбел в сердце Своём»

Позволено было в спасительный ковчег каждой твари по паре взять, сбежались все зверюшки к воротам, по мере прибытия в очередь выстраивались. Задал Господь задачу Ною, что взять он может только всех зверюшек подряд из одного интервала очереди (от и до), но только, чтоб каждого вида зверя чётное количество было из тех, что в интервал попали, а иначе погибнет всё человечество!

Ной, конечно, побольше зверюшек спасти хочет, а то есть будет нечего в столь долгом плавании, и обратился он к программистам современным...

Входные данные (ark.in):

Во входном файле зверюшки числами помечены, каждый тип зверюшки свой номер имеет. Очередь не более чем 20000 зверюшек (и соответственно целых положительных чисел в файле, записанных через пробел в порядке нахождения твари в очереди).

Наибольшее число, которым помечена зверюшка, соответствует числу разновидностей живности.

Выходные данные (ark.out):

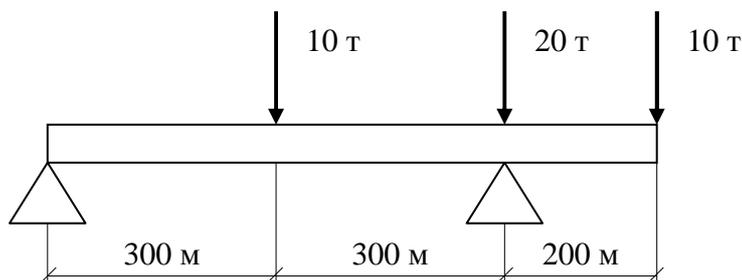
Вывести одно число – количество зверюшек, которых удастся спасти. Ну, а интервал Вы Ною скажите.

Пример:

ark.in	ark.out
1 2 2 1 3 2 1 2 1 3	10
1 2 1 3 1 3	4
1	0
1 2 4 4 2 3	4

Задача F**Опоры**

Строители нового моста через реку Великую обратились к вам за расчётами нагрузки на две опоры. Собственно им нужна программа, которая рассчитает давление на опоры при некотором приложении нагрузки (вектор нагрузки направлен вертикально вниз). На чертеже дан один из вариантов загрузки балки. При расчётах вес самой балки считать равным нулю, а место приложения нагрузки – точкой.

Входные данные (handhold.in):

В первой строке входного файла находится одно число N – количество нагрузок ($1 \leq N \leq 10$).

Затем на N строках дано по два целых числа: расстояние от левого края балки до места приложения нагрузки ($0 \leq L_i \leq 1000$) в метрах и сама нагрузка, выраженная в тоннах ($1 \leq P \leq 100$). И, наконец, в последней строке через пробел дано два различных целочисленных числа ($0 \leq L \leq 1000$) – расстояния до опор, отсчитанные от левого края балки.

Выходные данные (handhold.out):

Выведите два числа (по одному на каждой строке), величины давлений на опоры, сначала левой опоры, затем правой опоры. За положительное направление давления принять усилие, направленное вниз. Давление выведите точно с 6 знаками после запятой.

Пример:

handhold.in	handhold.out
3	1.666667
300 10	38.333333
600 20	
800 10	
0 600	

Задача G**Дерево файловой системы**

Вам поступил заказ от малоизвестных Вам людей, но щедро платящих, узнать глубину подчинённости в одной всем известной, но очень секретной организации. Благодаря таким данным преступники смогут оценить скорость принятия решения данной организации и использовать это в своих корыстных планах. Вам известно, что в данной организации существует строгая иерархия хранения информации на сервере. Вы смогли взломать сервер и выполнить команду `tree D:\Boss /F /A > tree.in`.

Файл получился очень большим и надо как-то автоматизировать процесс подсчёта глубины подчинённости. Boss имеет нулевой уровень подчинённости, то есть он не затрачивает время на принятие решения.

Входные данные (tree.in):

Содержит графическое отображение структуры папки D:/Boss (не более 113000 строк).

Выходные данные (tree.out):

Необходимо вывести одно число – максимальную глубину подчинённости.

Пример:

tree.in	tree.out
D:\BOSS	0
D:\BOSS 1.txt 2.txt 3.txt +---andrey +---tmp.txt 1.txt 2.txt 3.txt +---eugene 1.txt 2.txt +---sasha 1.txt \---roma 1.txt +---vovchik 1.txt \---dima 1.txt	2

Примечание: Имя файла не может содержать следующих знаков `\ / : * ? " < > |`