

Задача 1. Сложение столбиком**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

Однажды Волк поймал трех поросят и заставил их решать разные задачки. Одной из задач было сложение двух чисел (фантазия у Волка была слабовата). Пока числа были небольшие, поросята легко справлялись с задачей в уме. Когда числа стали побольше, они складывали их столбиком и проблем не возникало. Но когда Волк начал задавать очень длинные числа (20, 30, а то и 100 разрядов), поросята поняли, что без программы им не обойтись. Почувствуй себя четвертым поросенком – напиши нужную им программу.

Входные данные:

В первой строке входного файла записано число натуральное N , во второй – натуральное число M ($1 \leq N, M < 10^{100}$).

Выходные данные:

Выходной файл содержит в первой строке единственное число $S = N + M$.

Пример:

input.txt	output.txt
12	46
34	

Задача 2. Сумма цифр**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

Затем Волк решил проверить, как поросята умеют работать с цифрами. И задал задачу посчитать сумму первой и последней цифры числа n .

Входные данные:

В первой строке входного файла содержится единственное число n ($0 \leq n \leq 2 \times 10^9$).

Выходные данные:

Выходной файл содержит одно число – сумму первой и последней цифры числа n .

Пример:

input.txt	output.txt
15	6

Задача 3. Большая сумма цифр**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

Увидев, как легко поросята справились с предыдущей задачей, Волк понял, что даже самая обыкновенная деревенская свинья могла ее решить. Недолго думая, он решил самым простейшим образом изменить условие. А именно, складывать теперь он потребовал все цифры числа. Решив, что этого недостаточно, он также увеличил количество чисел – теперь надо было сложить цифры всех чисел от 0 до N .

Входные данные:В первой строке входного файла натуральное число N , $0 \leq N \leq 10^8$.**Выходные данные:**Выходной файл содержит одно число – сумму цифр всех натуральных чисел от 0 до N .**Пример:**

input.txt	output.txt
10	46

Задача 4. Скорее в укрытие
Входной файл: input.txt
Выходной файл: output.txt
Ограничение по времени: 1 секунда

Наконец-то поросятам удалось вырваться из волчьей норы. Теперь им надо как можно быстрее добраться до своего домика. Помогите поросятам выбрать самый короткий путь.

В лесу имеется N полянок и M тропинок между ними. Каждая тропинка соединяет две различные полянки. Две полянки могут быть соединены не более чем одной тропинкой. Бегать в лесу можно только по тропинкам.

Поросята выбегают с полянки, на которой находится волчья нора, и бегут до полянки, на которой находится их домик. Поросята давно живут в лесу и знают, сколько времени у них займет пробежка по каждой из тропинок.

Также поросята уверены, что духи леса на их стороне, и рассчитывают на их волшебную силу: один раз духи могут мгновенно перенести поросят по любой из тропинок (не зависимо от того, сколько времени заняло бы это перемещение у пеших поросят).

Входные данные:

Первая строка входного файла содержит числа N, M ($2 \leq N \leq 2000, 1 \leq M \leq 4000$). Следующие M строк содержат по три числа: B_i, E_i – номера полянок, которые соединяет i -я тропинка, и T_i – время, за которое поросята могут по ней пробежать ($1 \leq T_i \leq 1000$). В следующей (последней) строке два числа: номер полянки, на которой находится волчье логово, и номер полянки, на которой стоит дом поросят. Все числа во входном файле целые и в пределах одной строки разделены пробелами.

Выходные данные:

Одно целое число – минимальное время, за которое поросята доберутся до дома. Гарантируется, что решение задачи существует (нельзя же обрекать поросят на вечные скитания по лесу).

Пример:

input.txt	output.txt
7 10	4
1 2 2	
1 4 15	
2 3 3	
2 6 4	
3 7 3	
3 4 6	
3 6 3	
4 5 7	
4 7 2	
5 7 2	
1 5	

Задача 5. Грязевые ванны
Входной файл: input.txt
Выходной файл: output.txt
Ограничение по времени: 1 секунда

Поросятам удалось убежать от Волка, но снова попасться ему в лапы пороссятам крайне не хотелось, потому они договорились в духами леса, чтобы те выделили им безопасную область. Духи посоветались и выделили пороссятам треугольную область, в которой гарантировали им безопасность.

Поросята сначала очень обрадовались, а потом задумались. Раньше они любили поваляться в грязи. Но ведь могло так получиться, что ни одна лужа не попала в безопасную область. Теперь им нужно узнать, сколько в лесу осталось луж, в которых они могут чувствовать себя в полной безопасности.

Входные данные:

В первой строке шесть чисел через пробел – $x_1, y_1, x_2, y_2, x_3, y_3$ – координаты вершин треугольника.

Во второй строке число N ($1 \leq N \leq 1000$) – количество луж.

В третьей строке $2 \cdot N$ целых чисел через пробел – координаты луж x_i, y_i .

Все координаты по модулю не превышают 1000.

Выходные данные:

В первой строке одно целое число – количество луж, попавших в безопасный треугольник (лужи, попавшие на границы, также считаются попавшими в треугольник).

Пример:

input.txt	output.txt
0 0 3 3 3 0	1
2	
1 1 -1 -1	

Задача 6. Собираем желуди**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

Осенью поросята решили заготовить на зиму желудей. Они знают, что вдоль дороги через равные расстояния находятся дубовые рощи (будем считать расстояние между двумя соседними рощами единицей расстояния). Исходя из опыта, они знают, сколько корзин желудей соберут в каждой роще. Начинается дорога прямо от дома поросят, и ближайшая к дому (первая) роща находится от дома на расстоянии 1.

Чтобы не терять времени, поросята решили днем собирать желуди, а ночью ехать на своей поросычьей машине от одной рощи к другой. Так как это означало, что одному поросенку придется не спать целую ночь, было принято решение чередоваться: ни один поросенок не сидел за рулем машины две ночи подряд. Также следует учесть, что поросята водили машину с разными скоростями: осторожный Ниф-Ниф успевал за ночь проехать только до ближайшей рощи, спокойный Нуф-Нуф проезжал не останавливаясь одну рощу и доезжал до второй, а лихач Наф-Наф проезжал 3 единицы расстояния и довозил свинячье семейство до третьей рощи.

Важно заметить, что поросята всегда ехали только вперед и не возвращались назад.

Очевидно, что в сложившихся обстоятельствах все рощи посетить нельзя (иначе, Ниф-Ниф не будет спать очень много ночей подряд). В связи с этим у поросят возник вопрос, какое максимальное количество корзин с желудями они могут собрать, если все эти желуди нужно завезти в хранилище, расположенное в последней роще (конечно же, собрав желуди и в ней).

Входные данные:

Первая строка входного файла содержит число N ($1 \leq N \leq 50$) – количество дубовых рощ. В следующей строке через пробел числа K_i , $i = 1 \dots N$ в диапазоне от 1 до 100 каждое. K_i – количество корзин желудей, которое поросята соберут в i -ой роще.

Выходные данные:

Одно целое число – максимальное количество корзин желудей, которое поросята могут собрать, доехав до последней рощи (и собрав желуди в ней).

Пример:

input.txt	output.txt
4	55
15 20 30 10	

Задача 7. Накормить гостей**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

Однажды Волку пришло письмо. Оказалось, что к нему едут гости! А поскольку поросята убежали, волку нечем их кормить. Волк подумал и решил временно перейти на зайчатину. Но тут перед Волком встала новая проблема – нельзя, чтобы гостям досталось разное количество зайцев, иначе кто-то непременно обидится. «Значит, надо сделать так, чтобы количество зайцев делилось на число гостей,» – решил Волк. А лишних он решил съесть прямо сейчас (из-за возни с поросятами он весьма проголодался).

Волк начал считать зайцев и понял, что их расплодилось невероятно много. Он провел исследование заячьего рода и выяснил, что N лет назад появился один заяц (видимо, беременный). В первый год у нее родилось A_1 зайцев. На второй год родились новые зайцы, а старые годовалые зайцы умерли. При этом количество зайцев в лесу увеличилось в A_2 раз. То же самое повторялось каждый год. Изменялась только плодовитость зайцев A_i .

Помогите Волку посчитать, сколько зайцев ему надо съесть, что оставшееся число зайцев в лесу делилось на количество гостей.

Входные данные:

В первой строке два целых числа N ($1 \leq N \leq 10^4$) – количество лет и M ($1 \leq M \leq 10^9$) – количество гостей. Во второй строке N целых чисел A_i ($1 \leq A_i \leq 10^4$), где A_i – отношение количества зайцев в i -ый год к количеству зайцев в $i - 1$ -ый год (в «нулевой» год в лесу был один заяц).

Выходные данные:

Число R – минимальное количество зайцев, которое необходимо съесть волку, чтобы оставшееся количество зайцев делилось на число гостей.

Пример:

input.txt	output.txt
3 2 1 3 7	1

Задача 8. Сон**Входной файл:** input.txt**Выходной файл:** output.txt**Ограничение по времени:** 1 секунда

После приема гостей (см. предыдущую задачу) Волк лег спать. И приснился ему сон про зайцев. Снилось Волку, что как и в реальности, N лет назад в лесу появился один заяц. Но в первый год количество зайцев не изменилось. Во второй год зайцев стало в два раза больше. В третий – в три раза больше и так далее. Проснувшись, Волк решил узнать, пришлось бы ему опять-таки съесть некоторое количество зайцев в одиночку, чтобы оставшееся количество делилось ровно между всеми гостями, или нет. Помогите ему найти ответ на этот непростой вопрос.

Входные данные:

В первой строке два целых числа N ($1 \leq N \leq 10^{12}$) – количество лет и M ($1 \leq M \leq 10^{12}$) – количество гостей.

Выходные данные:

Выведите 'YES', если количество зайцев делится на количество гостей, и 'NO' в противном случае.

Пример:

input.txt	output.txt
3 6	YES

Задача 9. Участки

Входной файл: input.txt

Выходной файл: output.txt

Ограничение по времени: 1 секунда

Устав от борьбы с порослями, Волк занялся садоводством и огородничеством. Для этого он выделил себе прямоугольный участок земли, стороны которого шли с запада на восток и с севера на юг. К сожалению, на этом участке оказалось несколько бревен, которые, по странному стечению обстоятельств, также лежали либо горизонтально, либо вертикально. При этом, что не мало важно, никакие два бревна не пересекались и не лежали на одной прямой.

Убирать бревна Волку было лень, а красоты на участке хотелось. Потому он решил разбить свой участок на более мелкие непересекающиеся прямоугольные области так, чтобы выполнялись следующие условия:

1. Каждое бревно входит полностью в границы областей.
2. Число областей должно быть наименьшим.

Напишите программу, которая поможет Волку в его нелегком деле.

Входные данные:

В первой строке входного файла находятся координаты северо-восточного угла участка. Во второй строке указано количество бревен N ($1 \leq N \leq 1000$). Каждая из следующих N строк содержит описание одной из засек в виде четверки чисел x_1, y_1, x_2, y_2 , что соответствует отрезку с концевыми точками (x_1, y_1) и (x_2, y_2) . Все координаты являются неотрицательными целыми и не превосходят 1024. Числа в строке разделяются пробелом. Ни одно бревно не выходит за участок (но может лежать на его границе).

Выходные данные:

В первую строку выходного файла выведите количество областей M в найденном разбиении. В последующих M строках должны находиться целочисленные координаты углов для каждой из областей разбиения. Числа внутри строки разделять пробелами. Если вариантов разбиения с минимальным числом областей несколько, вывести любой из них.

Пример:

input.txt	output.txt
5 5	3
2	1 0 4 5
1 0 1 5	4 0 5 5
4 0 4 5	0 0 1 5