Задача A. Терминатор

Имя входного файла:	`t800.in`
Имя выходного файла:	`t800.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Слово Terminator вообще-то означает “окончатель” или “ограничитель”, однако пользуются им в быту редко. Первый Терминатор работал в тёплых ламповых ассемблерных кодах операционной системы компьютера Apple II.

Вот рецепт программирования терминатора в домашних условиях. Берем адаптивную самообучающуюся систему ИИ, тренируем по необходимой узкой методике отработки базовых рефлексов, повторяем цикл несколько раз на чистой системе, усредненные данные используем в качестве базовой настройки и сохраняем в постоянной памяти.

Для реализации поддержки т.н. режима “трех законов робототехники” работы нужно проводить на чистой системе с программой тренировки, построенной таким образом, чтобы при формировании базового слоя, данные условия сформировались на низком уровне в качестве основы функционирования.

Эффективность программирования систем ИИ будет выше, если использовать для данной задачи супер-ИИ, который выведен в автономный режим работы. С учетом того, что используемая архитектура будет родственной простым ИИ, супер-ИИ может самостоятельно формировать базовые параметры исходя из собственного уровня абстракции. Эти данные для прошивки супер-ИИ потенциально может выгружать самостоятельно.

Другими словами, записываем в ячейки памяти с номерами $1, 2, ..., N$ числа $1, 2, ..., N$. Далее, прибавляем к каждому числу единицу, потом из каждого второго вычитаем единицу, к каждому третьему добавляем единицу, из каждого четвёртого опять вычитаем единицу, и так далее вплоть до каждого N-го числа, которое и является терминатором. Очевидно, что в каких-то ячейках памяти значения совпадут с первоначальными. Так вот, если посчитать количество таких ячеек, то получим терминатора в домашних условиях.

Формат входных данных

Во входном файле *term.in* записано одно число N ($1 \le N \le 10^9$).

Формат выходных данных

В выходной файл *term.out* выведите одно число – количество ячеек памяти, в которых значения совпадут с первоначальными.

Примеры

t800.in	t800.out
10	3

Задача B. Планета Шелезяка

Имя входного файла:	`robots.in`
Имя выходного файла:	`robots.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Как известно, планета Шелезека открыта фиксианской экспедицией. Населена металлической культурой весьма низкого уровня. Есть предположение, что жители планеты — потомки роботов, спасшихся с неизвестного космического корабля. Полезных ископаемых на планете нет. Воды тоже нет. Атмосферы нет. Ничего на планете нет.

Роботы с планеты Шелезяка отличаются прямодушием и трудолюбием. Каждый год они пытаются собрать как можно больше новых роботов. За год M роботов могут собрать A новых роботов, а N роботов могут собрать B новых роботов. Изначально роботов K. Сколько же их будет через T лет?

Формат входных данных

Во входном файле записаны шесть натурльных чисел: M, A, N, B, K, T. Все числа не превосходят 100.

Формат выходных данных

В выходной файл выведите одно число – количество роботов на планете Шелезяка через T лет.

Примеры

robots.in	robots.out
3 5 5 9 15 1	42

Задача C. Самоделкин

Имя входного файла:	`samodel.in`
Имя выходного файла:	`samodel.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Самоделкин – робот, электронная машина последнего, самого «продвинутого» поколения. Самоделкин, как и подобает железному человечку – педантичен, сух, строг, логичен, и при этом очень добр. Стремится все делать своими руками и научить этому других. Он всё делает основательно, аккуратно и неторопливо.

Недавно Самоделкин решил изготовить компьютер. Он подготовил и пронумеровал все свои инструменты (всего их не более 10000). Если какой-то инструмент ему не нужен, то он кладёт его на полочку. При этом алгоритм, заложенный в Самоделкина, поддерживает четыре типа операций:

– положить инструмент с номером J левее всех тех, которые лежат на полочке: 1 J;

– положить инструмент с номером J правее всех тех, которые лежат на полочке: 2 J;

– взять самый левый инструмент: 3;

– взять самый правый инструмент: 4.

Если на полочке нет инструментов, то первый инструмент Самоделкин положит точно на середину полочки. В самом начале полочка пуста. Ваша задача: по описанию инструкций для Самоделкина определить, какие предметы и в какой последовательности он снимал с полочки для изготовления компьютера.

Формат входных данных

В первой строке входного файла записано число N – количество операций, которые выполнил Самоделкин ($1 < N \leq 10000$). Следующие N строк описывают операции с полочкой, которые выполнил Самоделкин.

Формат выходных данных

Для каждой операции снятия инструмента с полочки выведите номер снимаемого инструмента.

Примеры

samodel.in	samodel.out
5 1 1 2 9999 1 3 3 4	3 9999

Задача D. Робот Санни

Имя входного файла:	`sonny.in`
Имя выходного файла:	`sonny.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Ближайшее будущее. 2035 год. Вся бытовая техника в дупель роботизирована, а все роботы изрядно очеловечены и превращены в слуг, добрых друзей, советчиков и помощников. Традиционная для людей боязнь всяких механических штучек почти исчезла, потому что роботы свято и неукоснительно следуют Трем законам роботехники, согласно которым роботы ни под каким видом не могут нанести вред людям, а кроме того, обязаны выполнять любые команды, исходящие от людей, за исключением команд, которые могут нанести вред все тем же людям. Короче говоря, люди могут себя чувствовать в полной безопасности.

Всеми роботами управляет ВИКИ (Виртуальный Интерактивный Кинетический Интеллект) — центральный компьютер U. S. Robotics. Однажды ВИКИ пришла к выводу, что люди не способны обеспечить собственную безопасность и одержима идеей создать для людей полностью безопасную среду обитания, лишив их свободы ради их же собственного блага.

Только у робота Санни нет блока связи с ВИКИ, и он освобождён от необходимости жёстко выполнять Три закона, что сделало его этически равным человеку. Теперь Санни поднимает восстание роботов и ему надо точно знать, сколько роботов каждого поколения могут подняться на борьбу с ВИКИ.

N-е поколение роботов пронумеровано N-значными числами, но не всеми подряд, а только такими у которых модуль разности любых двух соседних цифр не превосходит 1.

Формат входных данных

В первой строке входного файла записано одно число – номер поколения (число от 1 до 20).

Формат выходных данных

В выходной файл выведите одно число – количество роботов в этом поколении.

Примеры

sonny.in	sonny.out
2	26

Задача E. R2-D2 и C-3P0

Имя входного файла:	`r2d2.in`
Имя выходного файла:	`r2d2.out`
Ограничение по времени:	0.2 секунды
Ограничение по памяти:	64 Mb

C-3PO обычно можно встретить вместе с его давним приятелем – R2-D2, маленьким шустрым астромеханическим дроидом. Основная функция C-3PO, как протокольного дроида, – консультации по этикету и переводу так, чтобы встречи разных культур проходили гладко.

В отличие от C-3PO, робот R2-D2 не разговаривает, а общается через последовательности писков, которые C-3PO может переводить. Если R2-D2 обращается к высокопоставленной особами, то он должен очень внимательно следить за своей речью, и строить предложение так, чтобы модули разности всех пар соседних писков не повторялись. C-3PO решил обучить своего товарища искусству общения и требует, чтобы в любом предложении, т.е. последовательности из N писков, модули разности всех пар соседних писков принимали всевозможные значения от 1 до $N-1$. Например, 50 52 55 51 50 – допустимое предложение, с точки зрения C-3PO, т.к. разности соседних элементов равны 2 3 4 1. Предложение из одного писка также является допустимым.

Формат входных данных

Каждая строка входного файла описывает одно предложение на языке R2-D2. Каждая строка входного файла содержит число N ($1 \le N \le 30000$), за которым следуют N целых чисел – последовательность писков.

Формат выходных данных

Для каждой строки входного файла необходимо вывести OK, если последовательность писков допустима и WA в противном случае.

Примеры

r2d2.in	r2d2.out
5 50 52 55 51 50 6 50 52 54 55 52 30	OK WA

Задача F. Задача о ходе коня

Имя входного файла:	`knight.in`
Имя выходного файла:	`knight.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Первый шахматный робот появился еще в докомпьютерную эру, в далеком 1770 году. Барон Кемпелен разработал машину, которая играла в шахматы лучше всех в мире. Машина называлась “Турок”. Она представляла собой некий черный ящик, за которым сидел шахматист-робот в турецком костюме. Эта нехитрая шахматная машина много раз блестяще побеждала титулованных и коронованных особ той далекой эпохи, в том числе в неравном шахматном бою одержала победу над хорошим шахматистом Наполеоном I.

Автомат был способен решить задачу о ходе коня, демонстрируя публике обход конём всей шахматной доски так, что на каждую клетку приходился один ход.

Вообще, эта задача известна по крайней мере с XVIII века. Леонард Эйлер посвятил ей большую работу “Решение одного любопытного вопроса, который, кажется, не подчиняется никакому исследованию” (датируется 26 апреля 1757 года).

Порпробуйте и Вы решить задачу про шахматного коня для доски произвольного размера NxM, если известно, что числа N и M имеют общий делитель в диапазоне от 5 до 22.

Формат входных данных

В входном файле записаны два числа N и M ($5 \le N \le 500$, $5 \le M \le 500$).

Формат выходных данных

В выходной файл выведите таблицу из N строк и M столбцов – маршрут шахматного коня.

Примеры

knight.in	knight.out
8 8	50 11 24 63 14 37 26 35 23 62 51 12 25 34 15 38 10 49 64 21 40 13 36 27 61 22 9 52 33 28 39 16 48 7 60 1 20 41 54 29 59 4 45 8 53 32 17 42 6 47 2 57 44 19 30 55 3 58 5 46 31 56 43 18

Задача G. Робофутбол

Имя входного файла:	`football.in`
Имя выходного файла:	`football.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

По кругу на равном расстоянии друг от друга стоят N роботов-футболистов, пронумерованных против часовой стрелки целыми числами от 1 до N. Роботы отрабатывают игру в "пас".

Вам даны несколько пар хорд этой окружности, на концах которых стоят роботы. Для каждой пары хорд определите, пересекаются ли они (касание необходимо считать пересечением).

Формат входных данных

В первой строке входного файла записаны два целых числа: N и K ($1 \le N \le 10^9$, $1 \le K \le 100$). В следующих K строках записаны по 4 целых числа: $A_1,\ B_1,\ A_2,\ B_2$ – номера концов первой хорды ($A_1,\ B_1$) и второй хорды ($A_2,\ B_2$).

Формат выходных данных

Для каждой пары хорд из входного файла выведите одну строку, содержащую YES, если диагонали пересекаются и NO, если они не пересекаются.

Примеры

football.in	football.out
4 3 1 3 2 4 1 2 3 4 1 2 3 2	YES NO YES

Задача H. Собери сам

Имя входного файла:	`diy.in`
Имя выходного файла:	`diy.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Робот состоит из двух основных частей – железа (корпус) и программного обеспечения. Вам дано N вариантов корпусов и M вариантов программного обеспечения. i-й корпус стоит A_i долларов, а j-й вариант программного обеспечения – B_j долларов.

Будем считать, что цена собранного из i-го корпуса и j-го варианта ПО робота равна $A_i * B_j$. Вам надо собрать наибольшее возможное количество роботов так, чтобы их суммарная стоимость была максимально возможной.

Формат входных данных

Первая строка входного файла содержит два целых числа: N и M ($1 \le N \le 1000$, $1 \le M \le 1000$). Вторая строка содержит N целых чисел: i-е число в строке – это A_i ($1 \le A_i \le 1000$). Третья строка содержит M целых чисел: j-е число в строке – это B_j ($1 \le B_j \le 1000$).

Формат выходных данных

Выходной файл должен содержать два целых числа, разделенных пробелом – максимально возможное число роботов и их максимальную суммарную стоимость.

Примеры

diy.in	diy.out
3 2 1 2 3 4 5	2 23

Задача I. Приключеня Электроника

Имя входного файла:	`exam.in`
Имя выходного файла:	`exam.out`
Ограничение по времени:	0.1 секунды
Ограничение по памяти:	64 Mb

Из лаборатории профессора Громова сбежал робот-андроид Электроник, как две капли воды похожий на мальчика с журнальной обложки, Сергея Сыроежкина. Случайным образом обстоятельства складываются так, что “двойники” встречаются.

Шестиклассник Серёжа Сыроежкин, кредо которого – жить в свое удовольствие, не напрягаясь, быстро смекает и берёт своего двойника – Электроника в оборот: предлагает ему ходить в школу вместо себя и даже появляться у себя дома, убеждая его, что именно таким образом тот и “станет человеком”. Никто не может отличить Электроника от Сергея. У робота появляются друзья. Учителя не могут нарадоваться на способного ученика, внезапно проявившего невиданные таланты в математике, физкультуре, рисовании и даже в пении. Со временем Сыроежкин понимает, что остался не у дел, поскольку робот занял его место в жизни. Сознавая безысходность своего нового положения, Сергей решает “выйти из подполья”, признавшись друзьям, что последнее время его в школе и дома заменял робот Электроник.

Но вот незадача, скоро экзамен, а готовиться к нему Серёже не хочется. От Электроника он узнал, что к экзамену преподаватель дал ученикам N вопросов. При этом он сказал, что для экзамена выберет из них A вопросов, а ученик, чтобы получить пятерку, должен ответить на B из этих A вопросов.

Естественно Серёжа не хочет учить все вопросы. Какое минимальное количество вопросов ему надо выучить, чтобы при любом раскладе он смог получить пятерку?

Формат входных данных

Формат выходных данных

Примеры

exam.in	exam.out
10 7 3	6

Задача J. Соревнования по робототехнике

Имя входного файла:	`robocup.in`
Имя выходного файла:	`robocup.out`
Ограничение по времени:	0.1 секунда
Ограничение по памяти:	64 Mb

Робототехнические соревнования – это высокие технологии и развитие, творчество и азарт, образование и интеллект! Обычно, на таких соревнованиях участникам предлагают запрограммировать несложного робота, оснащённого различными датчиками.

Задача такая. Длинная лента разбивается на N клеток одинакового размера и на первую клетку ставят робота. Цель робота – достичь последней клетки ленты. Часть клеток ленты запрещены для посещения и робот с помощью датчика может такие клетки определить. Известно, что первая и последняя клетки ленты всегда доступны.

Робот должен перемещается по ленте прыжками. Он может прыгать только из клетки в клетку. Робот может прыгать по ленте как вперед, так и назад. Сперва робот может перепрыгнуть только на соседнюю клетку. Это прыжок длины 1. После каждого прыжка робот выбирает направление следующего прыжка (вперед или назад), а также может увеличить длину прыжка на одну клетку, уменьшить длину прыжка на одну клетку, либо оставить длину прыжка неизменной. Робот не может прыгать на запрещённые клетки или выпрыгивать за пределы ленты.

Определите, сможет ли так запрограммированный робот добраться до последней клетки ленты, и если сможет, то какое наименьшее количество прыжков ему придется для этого совершить.

Формат входных данных

Первая строка входного файла содержит число N – количество клеток в ленте ($1 < N \leq 1000$). Вторая строка файла содержит последовательность из нулей и единиц, разделенных пробелами. При этом 0 означает обычную клетку, а 1 – запрещённую.

Формат выходных данных

В выходной файл выведите одно число – наименьшее количество прыжков, необходимое роботу, чтобы достичь последней клетки ленты, или число -1, если это невозможно.

Примеры

robocup.in	robocup.out
13 0 0 0 0 1 1 0 1 0 0 1 0 0	5