Задача A. Анаграммы-2

Имя входного файла:	`anagrams2.in`
Имя выходного файла:	`anagrams2.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Недавно Человек-Невидимка от нечего делать прогуливался по крышам домов и случайно подслушал интересный разговор, доносящийся из открытого окна последнего этажа. Разговаривали два человека, одного из которых звали «Нолик», а второго — «Симка». «`Странные имена`», — подумал Человек-Невидимка. Но для него это было неважно, намного интереснее была тема разговора — это было что-то, связанное с программированием, а он никогда не мог пройти мимо такого соблазна.

Внимательно все послушав, Человек-Невидимка понял, что суть задачи, которую обсуждали эти два странных человека, состоит в следующем: по данному массиву-шаблону и массиву-тексту надо было понять, существует ли такой подотрезок текста, совпадающий с массивом-шаблоном как анаграмма. Под анаграммами в данном случае понимались два слова, в которых можно как-то переставить буквы, чтобы они стали одинаковыми. Оценив задачу, Человек-Невидимка понял, что она для него слишком простая, поэтому он решил усложнить ее. После некоторых раздумий, ему в голову пришла следующая ее модификация: по данным двум массивам требовалось найти такое максимальное число k, что в первом и втором массивах существуют подотрезки длиной k, совпадающие как анаграммы. Но эта задача уже оказалась Человеку-Невидимке не по силам, поэтому он попросил у вас помощи в решении этой задачи.

Формат входного файла

В первой строке входного файла дано число n ($1 \le n \le 1\,000$) — длина первого массива.

Во второй строке через пробел заданы n чисел a_i ($1 \le a_i \le 100\,000$) — первый массив.

В третьей строке дано число m ($1 \le m \le 1\,000$) — длина второго массива.

В четвертой строке через пробел заданы m чисел b_i ($1 \le b_i \le 100\,000$) — второй массив.

Формат выходного файла

В единственной строке выходного файла выведите три неотрицательных числа k, i, j — максимальная длина подотрезков, совпадающих как анаграммы, а также начало отрезка в первом массиве и во втором соответственно. Если максимальная длина подотрезка равна 0, следующие два числа в выходном файле должны равняться -1.

Примеры

anagrams2.in	anagrams2.out
3 1 2 3 3 3 2 1	3 1 1
3 1 2 3 3 4 5 6	0 -1 -1

Задача B. Хэллоуин

Имя входного файла:	`bipartite.in`
Имя выходного файла:	`bipartite.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В этом году на Хэллоуин, как обычно, монстры будут пугать людей. В этом году граф Дракула взял на себя ответственность все спланировать. У каждого монстра он узнал, какие дома тот хочет «навестить», соединил линиями на листке соответствуюещего монстра с соответствующими домами, а после этого оставил листок с записями на своем столе в кабинете и ушел по делам.

В его отсутвие Квазимодо зашел в кабинет и увидел на столе интересный листок какими-то линиями — листок, оставленный Дракулой. Недолго думая, он нарисовал еще несколько линий на нем, а потом положил лист на место и ушел. Вернувшись, граф Дракула не мог не заметить изменения в листе. Кроме того, он осознал, что после добавлений Квазимодо теперь не ясно, где на листке отмечены дома, а где монстры. И следовательно, Хэллоуин на грани срыва.

Теперь Дракула просит вас попробовать вернуть лист с линиями в начальное состояние, то есть стереть с него несколько линий так, чтобы после этого все объекты на листике можно было разбить на две группы — дома и монстры, чтобы линии проходили только между этими группами. Он не верит, что Квазимодо мог нарисовать очень много линий, поэтому он попросил вас удалить у каждого объекта не более половины исходящих из него линий. Помогите Дракуле привести лист к корректному состоянию или скажите, что его предположение неверно.

Формат входного файла

В первой строке входного файла содержится два числа n, m ($1 \le n, m \le 2\,000$) — количество объектов на листке и линий между ними соответственно.

В следующих m строках содержится описание двусторонних линий, соединяющих объекты. В $i+1$-й строке входного файла содержится описание i-й линии. Описание состоит из двух чисел u, v ($1 \le u, v \le n, u \ne v$) — номера объектов, соединенных линией. Гарантируется, что между одной парой объектов проведено не более одной линии.

Формат выходного файла

В первой строке выходного файла выведите количество удаленных линий.

Во второй строке выведите номера удаленных линий. Линии нумеруются с 1 в порядке следования во входном файле. После удаления количество исходящих линий у каждой вершины должно уменьшиться не более чем вдвое.

Примеры

bipartite.in	bipartite.out
3 3 1 2 2 3 3 1	1 1
4 5 1 2 1 3 1 4 2 3 3 4	2 1 5

Задача C. Соревнование

Имя входного файла:	`competition.in`
Имя выходного файла:	`competition.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Чтобы развлечь своих постояльцев, Граф Дракула решил организовать соревнование. В нем приняли участие все n монстров, отдыхающих в отеле. Граф Дракула придумал k дисциплин, в каждой из которых участники могли набрать от 0 до 10 баллов. Считается, что участник x выступил лучше, чем участник y, если больше чем в половине дисциплин участник x набрал строго больше баллов, чем участник y.

К сожалению, затея Графа Дракулы оказалась не такой хорошей, как показалось в начале. Все участники перессорились друг с другом, ведь каждый хотел победить! Чтобы все были довольны, Граф Дракула решил незаметно подтасовать результаты так, чтобы участник с номером i выступил формально лучше, чем участник с номером $(n+i-1) \mod n$ и хуже, чем участник с номером $(i+1) \mod n$, если нумерация участников начинается с 0.

Помогите Графу Дракуле — напишите программу, которая генерирует для каждого участника его результат в каждой из k дисциплин.

Формат входного файла

В единственной строке входного файла даны два числа n, k ($4 \le n, k \le 100$) — количество участников и дисциплин.

Формат выходного файла

Выведите n строк. В i-й строке выходного файла выведите результат i-о участника в каждой из k дисциплин.

Пример

competition.in	competition.out
7 5	4 2 0 4 1 0 3 2 1 3 1 4 2 2 0 2 3 3 4 1 3 4 1 0 2 1 0 2 2 3 2 0 3 3 1

Задача D. Игра

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Франкенштейн и Марта играют в игру. В этой игре каждый из игроков получает по одному числу — игрок, который ходит первым, берет себе число A, а игрок который ходит вторым, получает число B. Кроме того, в правилах этой игры указано число C. На своем ходу игрок может взять любой делитель своего числа, не превышающий C и больший единицы, и поделить свое число на этот делитель. Есть лишь одно условие: наибольший общий делитель числа, которое останется у этого игрока и числа, которое есть в данный момент у другого игрока, должен быть больше единицы. Проигрывает тот, кто не может сделать ход.

Как джентльмен, Франкенштейн уступает Марте право выбора хода. Хитрая Марта поняла, что по числам A, B, C можно понять, который из игроков выиграет при оптимальной игре, но не смогла определить, кто именно побеждает. Подскажите ей, кто выиграет.

Формат входного файла

В первой строке даны три целых числа A, B, C ($2 \le A, B, C \le 10^9$) — числа, которые есть у игроков, и ограничение на делитель.

Формат выходного файла

Выведите «First», если выиграет первый игрок, и «Second», если выиграет второй игрок.

Пример

game.in	game.out
20 15 5	First
12 16 5	Second

Задача E. Мэйвис и Дракула

Имя входного файла:	`lcm.in`
Имя выходного файла:	`lcm.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сегодня один из тех вечеров, когда Мэйвис и Дракула проводят время вместе. Дракула решил поиграть с дочкой в игру. Она, конечно же, как вы уже наверное догадались, имеет математическую основу.

Состоит эта игра в следующем: Дракула говорит Мэйвис натуральные числа A и B, после чего Мэйвис должна найти такое натуральное число x, не превосходящее B, что $lcm(A, x)$ максимально среди всех натуральных чисел, не превосходящих B. Дракула хочет сыграть с дочкой T раундов.

Мэйвис достаточно быстро поднадоела эта математическая игра, и она решила попросить вас написать программу, которая по заданным A и B будет находить число наибольшее значение lcm, которого можно достигнуть.

Формат входного файла

В первой строке содержится одно натуральное число T ($1 \le T \le 10^5$) — количество раундов в игре. В следующих T строках содержатся числа A и B ($1 \le A, B \le 3 \times 10^9$) для каждого раунда.

Формат выходного файла

В T строках должны содержаться ответы для раундов — наибольшее значение lcm, которого можно достигнуть в этом раунде.

Пример

lcm.in	lcm.out
6 10 20 20 13 8 4 17 34 4 11 18 20	190 260 24 561 44 342

Задача F. Прямая на плоскости

Имя входного файла:	`stdin`
Имя выходного файла:	`stdout`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Даже монстрам нельзя забывать про обучение. Поэтому граф Дракула в прошлом году организовал школу для самых маленьких монстров, где их учат основам поведения в обществе, пугания людей, а также математике — ну а как же без нее. И вот завтра Дракула как раз собрался провести очередной урок по построению графиков функций, а именно — прямых. Он хочет рассказать, что любую не вертикальную прямую на плоскости можно задать уравнением $y = kx + b$. Но этого явно мало, поэтому вторую половину урока Дракула решил уделить играм для улучшения понимая материала. Из всех вариантов он выбрал следующий:

Маленькие монстрики задумывают два целых числа k, b ($|k|, |b| \le 1\,000$, $k \neq 0$) — коэффициенты уравнения $kx + b = 0$, но Дракуле они их не сообщают. После этого граф должен их отгадать. За один ход он может написать на бумажке два целых числа k_0, b_0 ($|k_0|, |b_0| \le 5\,000$), после этого монстрики скажет ему, сколько корней у текущего уравнения $kx + b = 0$, и если корень ровно один, они также скажут его знак. После этого текущий коэффициент k заменяется на $k + k_0$, а текущий коэффициент b на $b + b_0$.

Дракула не хочет упасть в глазах своих учеников, поэтому просит вас быть на связи и помогать ему выбирать нужные числа k_0, b_0. Не откажите самому известному вампиру, помогите ему найти задуманные монстриками коэффициенты не более чем за 50 ходов.

Протокол взаимодействия с программой жюри:

Во время взаимодействия вашей программы с программой жюри несколько раз повторяются следующие действия:

- ваша программа сообщает программе жюри три целых числа $type$, k_0, b_0 ($|k_0|, |b_0| \le 5\,000$). $type$ равно 0, если вы хотите получить от жюри информацию о текущем уравнии и 1, если следующие два числа — ответ на задачу. В случае $type = 1$ после вывода вам необходимо сразу же завершить программу.
- программа жюри сообщает вашей программе:
  - «`-1`», если у текущего уравнения $kx + b = 0$ бесконечное число корней
  - «`0`», если у текущего уравнения нет корней
  - «`1 1`», если у текущего уравнения ровно один корень, и он положительный
  - «`1 -1`», если у текущего уравнения ровно один корень, и он не положительный
- в случае, если в вашем запросе $type = 0$, описанные действия начинают повторяться сначала

Пример

stdin	stdout

`1 -1`
`1 -1`
`-1`
	`0 1 0`
	`0 -2 -1`
	`0 0 0`
	`1 1 1`

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача G. Мэйвис в школе

Имя входного файла:	`maxxor.in`
Имя выходного файла:	`maxxor.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Когда-то давным давно Мэйвис училась в вампирской школе. На уроках информатики Мэйвис решала задачки на различные темы. Одна из ее любимых тем — это массивы.

И вот наступил важный день: в вампирской школе контрольная по этой теме. На контрольной ей досталась следующая задача: дан массив a_i, нужно найти в нем отрезок $[l..r]$ с максимальной величиниой $max(l..r) \times xor(l..r)$.

$max(l..r)$ — это максимальное число на отрезке $[l..r]$, а $xor(l..r)$ — это $\oplus$-сумма чисел на этом отрезке.

Помогите ей решить эту задачу, ведь она очень не хочет огорчать Дракулу.

Формат входного файла

В первой строке находятся натуральное число n ($1 \le n \le 10^5$), в следующей строке находятся n чисел a_i ($1 \le a_i \le 10^6$) — массив, который есть у Мэйвис.

Формат выходного файла

В первой строке выведите целое число — наибольшее произведение $max(l..r) \times xor(l..r)$, которое можно найти в массиве.

Пример

maxxor.in	maxxor.out
4 1 2 10 7	150
6 2 3 5 3 6 8	112

Комментарий

В первом примере ответом является отрезок [2..4].

Во втором примере ответом является отрезок [5..6].

Задача H. Задачка о строке

Имя входного файла:	`minimal.in`
Имя выходного файла:	`minimal.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Джонатан знает много забавных вещей из большого мира. Сегодня он задал Мэйвис следующую задачку (наверное, он знает ее с какой-нибудь олимпиады по информатике, а может и еще откуда-нибудь):

Есть строка, а также указатель, который изначально указывает на первый символ строки. Доступны две операции:

1. Дописать символ на текущей позиции в конец ответа, если раньше символ с этой позиции не был взят (при этом старый символ остается на этой позиции).
2. Передвинуть указатель вправо на один символ. Если текущий символ последний, то указатель передвигается на первый символ строки.

После выполнения некоторого количества операций все символы должны быть взяты, а символы в строке ответа должны быть упорядочены по неубыванию. Например, если у нас есть строка hello, то мы можем выполнить следующие операции:

1. Вторая операция. Сдвигаем указатель на символ «e».
2. Первая операция. Берем символ «e».
3. Вторая операция. Указываем на символ «l».
4. Вторая операция. Указываем на символ «l».
5. Вторая операция. Указываем на символ «o».
6. Вторая операция. Теперь мы указываем на первый символ строки — «h».
7. Первая операция. Берем «h», теперь строка ответа равна «eh».
8. Вторая операция. Мы указываем на уже взятый символ «e».
9. Вторая операция.
10. Первая операция. Ответ «ehl».
11. Вторая операция.
12. Первая операция. Ответ «ehll».
13. Вторая операция.
14. Первая операция. Ответ «ehllo».

Итого, мы выполнили 5 операций первого типа и 9 операций второго типа.

Вам предстоит решить немного модифицированную версию этой задачи в общем случае.

Формат входного файла

В первой строке содержится строка s ($1 \le |s| \le 10^5$) — строка состоящая из строчных латинских букв. Во второй строке находится число m ($1 \le m \le 10^5$) — количество запросов. В каждой из следующих m строк находится по два числа l_i и r_i — границы очередного запроса.

Формат выходного файла

Для каждого запроса выведите количество операций второго типа, которые необходимо выполнить, чтобы решить задачу на подстроке s с l_i-й позиции до r_i-й включительно.

Пример

minimal.in	minimal.out
hello 3 1 5 1 2 2 5	9 2 3
fedcba 1 1 6	30

Задача I. Очередь

Имя входного файла:	`queue.in`
Имя выходного файла:	`queue.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Совсем недавно вышел новый фильм — «Люди на каникулах». По этому поводу Мэйвис решила сходить в кинотеатр.

Оказалость, что Мэйвис далеко не единственная, кто хочет посмотреть этот фильм. За билетами была огромная очередь. Мэйвис решила, что посмотрит кино в следующий раз, а пока она просто понаблюдает за очередью.

Билеты на фильм продаются в павильоне, в котором одновременно могут находиться не более m монстров. Монстры внутри павильона обслуживаются в порядке очереди. То есть, если i-й монстр зашел в павильон раньше j-о, то монстр с номером i купит билет раньше монстра с номером j. Монстры очень привередливы, поэтому на выбор билета у i-о монстра уходит h_i минут.

Если i-й монстр подходит к павильону в момент времени t_i и в павильоне в это время ровно m монстров, то он уходит и возвращается через k минут, то есть к моменту времени t_i+k. Иначе монстр заходит в павильон. Монстры очень упорные, поэтому каждый монстр будет возвращаться, пока не купит билет. Если несколько монстров подходят к павильону в одно и то же время, то сначала пытается зайти монстр с меньшим номером. Если i-й монстр в павильоне завершил покупку, и в это же время к павильону подходит j-й монстр, то сначала i-й выходит, а потом j-й пытается войти.

Мэйвис стало интересно, в какой момент времени каждый монстр купит билет. Помогите ей удовлетворить любопытство!

Формат входного файла

В первой строке входного файла даны три целых числа n, m, k ($1 \le n, m \le 10^5, 1 \le k \le 10^9$) — количество монстров, максимальное число монстров в павильоне и время, на которое уходит не поместившийся монстр.

В следующих n строках дано описание монстров.

В $i+1$-й строке даны два целых числа t_i, h_i ($1 \le t_i, h_i \le 10^9$) — время, в которое приходит i-й монстр в первый раз и время, которое он тратит на покупку билета.

Гаранитируется, что $t_i \le t_{i+1}$.

Монстры пронумерованы в порядке, в котором они идут во входных данных.

Формат выходного файла

Выведите n строк. В i-й строке выведите время, в которое i-й монстр купит билет.

Примеры

queue.in	queue.out
7 3 4 0 5 7 3 8 7 9 6 10 10 13 10 14 10	5 10 17 23 33 43 53
5 1 6 0 7 1 2 1 6 1 7 6 2	7 9 25 32 14
3 3 100 0 1 0 1 0 1	1 2 3

Задача J. Заклинания

Имя входного файла:	`strings.in`
Имя выходного файла:	`strings.out`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Мэйвис хочет применить очень сильное заклинание, чтобы вернуть Джонатана. У нее есть несколько обычных заклинаний, которые она хочет скрестить, чтобы получить сильное. Все заклинания представляют собой непустые строки из строчных латинских букв. У Мэйвис есть k обычных заклинаний a_i. Она также знает заклинание t, которое она хочет получить в результате.

Процесс скрещивания происходит следующим образом: изначально у Мэйвис есть пустая строка s, Мэйвис производит несколько этапов скрещиваения: во время каждого этапа она выбирает номер обычного заклинания i и делает одно из действий

- $s = s + a_i$ — строки конкатенируются;
- $s = s \times a_i$ — строки умножаются: после каждого символа строки s записывается строка a_i.

Обратите внимание, что операция $\times$ не коммутативна.

В результате скрещивания Мэйвис хочет получить строку t. Помогите ей определить, какое минимальное количество этапов потребуется для этого.

Формат входного файла

В первой строке содержится строка t ($1 \le |t| \le 5 \times 10^4$) — заклинание, которое хочет получить Мэйвис, в следующей строке находится число k — количество обычных заклинаний, имеющихся у нее, в следующих k ($1 \le k \le 1000$) строках находятся обычные заклинания a_i ($1 \le |a_i| \le 100$) по одному в строке.

Все строки непустые и состоят из строчных латинских букв.

Формат выходного файла

В единственной строке выведите минимальное количество этапов, которое потребуется Мэйвис для получения строки t, либо -1, если это сделать невозможно.

Пример

strings.in	strings.out
abbabbabb 3 aa bb a	3
arbrcrxryrzrr 3 abc xyz r	4
dabcba 4 b ab c d	-1