

Задача А. 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 соответствующего монстра с соответствующими домами, а после этого оставил листок с записями на своем столе в кабинете и ушел по делам.

В его отсутствие 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 ($1 \leq n, m \leq 2000$) — 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 ($1 \leq u, v \leq n, u \neq v$) — 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 В. 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 $(n+i-1) \bmod n$ и хуже, чем участник с номером $(i+1) \bmod n$, 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 ($4 \leq n, k \leq 100$) — 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

| <code>competition.in</code> | <code>competition.out</code> |
|-----------------------------|---|
| 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 С. Чемпионат

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

Граф Дракула решил устроить среди монстров чемпионат по устрашению.

Для этого он разбил всех монстров на две команды, в первой команде — n монстров, а во второй — m монстров. Чтобы избежать жульничества, для каждого монстра Дракула разузнал и записал на листок с какими монстрами из команды соперника этот монстр знаком. Эта информация должна была помочь ему следить за участниками во время соревнования, но, к сожалению, Дракула потерял листок. Он запомнил для каждого монстра только количество его знакомых в другой команде.

Так как Граф в ярости, то вам нужно поскорее выяснить кто с кем знаком.

Формат входного файла

В первой строке входного файла через пробел записаны числа n и m ($1 \leq n, m \leq 100$). Во второй строке через пробел записано n чисел a_i ($0 \leq a_i \leq 100$) — количество знакомых с i -м членом первой команды членов второй команды. В третьей строке через пробел записано m чисел b_i ($0 \leq b_i \leq 100$) — количество знакомых с i -м членом второй команды членов первой команды.

Формат выходного файла

Выведите YES, если решение существует, в противном случае выведите NO. Если решение существует, то во второй строке выведите количество пар монстров, где первый монстр из первой команды, а второй — из второй, и они знакомы. Далее выведите все эти пары монстров по одной в строке в любом порядке.

Пример

| <code>friends.in</code> | <code>friends.out</code> |
|-------------------------|---|
| 3 3 1 0 2 1 1 1 | YES 3 1 1 3 2 3 3 |
| 3 3 3 3 3 3 3 3 | YES 9 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3 |

Задача D. 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 $\text{lcm}(A, x)$ максимально среди всех натуральных чисел, не превосходящих B . Дракула хочет сыграть с дочкой T раундов.

Мэйвис достаточно быстро поднаделась 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 ($1 \leq T \leq 10^5$) — количество раундов в игре. В следующих T строках содержатся числа A и B ($1 \leq A, B \leq 3 \times 10^9$) для каждого раунда.

Формат выходного файла

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

Пример

| <code>lcm.in</code> | <code>lcm.out</code> |
|---------------------|----------------------|
| 6 | 190 |
| 10 20 | 260 |
| 20 13 | 24 |
| 8 4 | 561 |
| 17 34 | 44 |
| 4 11 | 342 |
| 18 20 | |

Задача Е. 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 понимания материала. Из всех вариантов он выбрал следующий:

Маленькие монстрики задумывают два целых числа k, b ($|k|, |b| \leq 1\,000, k \neq 0$) — 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 ($|k_0|, |b_0| \leq 5\,000$), после этого монстрики скажут 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 $type, k_0, b_0$ ($|k_0|, |b_0| \leq 5\,000$). $type$ равно 0, если вы хотите получить от жюри информацию о текущем уравнении 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 0 |
| 1 -1 | 0 1 |
| 1 -1 | -2 -2 |
| -1 | 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 F. 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 ($1 \leq |s| \leq 10^5$) — строка состоящая из строчных латинских букв. Во второй строке находится число m ($1 \leq m \leq 10^5$) — 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 | 9 |
| 3 | 2 |
| 1 5 | 3 |
| 1 2 | |
| 2 5 | |
| fedcba | 30 |
| 1 | |
| 1 6 | |

Задача G. В тесноте, да не в обиде

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

В последнее время популярность отеля резко возрасла, и комнат стало не хватать. Граф Дракула решил поделить одну из больших старых комнат на три новые.

Старая комната представляет собой прямоугольник длиной n и шириной m . Граф хочет разделить ее на три новых прямоугольных комнаты, так чтобы сумма периметров новых комнат была равна p .

Помогите Дракуле определить можно ли разделить таким образом комнату.

Формат входного файла

В единственной строке через пробел записаны три числа n , m и p ($1 \leq n, m, p \leq 10^5$).

Формат выходного файла

Выведите YES, если комнату можно разделить на три новых, в противном случае выведите NO.

Пример

| <code>rectangle.in</code> | <code>rectangle.out</code> |
|---------------------------|----------------------------|
| 2 2 14 | YES |
| 2 2 15 | NO |

Задача Н. Переходы переходов

| | |
|-------------------------|--------------------------|
| Имя входного файла: | <code>streets.in</code> |
| Имя выходного файла: | <code>streets.out</code> |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Чтобы добраться до офиса городского представительства отеля «Трансильвания» Желе нужно преодолеть непростой путь, а именно пройти по длинному проспекту с множеством перекрестков в том числе Т-образных (у которых отсутствует один боковой проезд).

Чтобы Желе было легче ориентироваться, Зомби выдали ему схематическую карту, на которой последовательно пронумерованы все дома. При этом номера противоположных домов, стоящих на разных сторонах проспекта, совпадают. Также на карте отмечены все переходы дорог, которые есть на проспекте. Переходы бывают трех типов:

- Вдоль проспекта на левой его стороне. Переход расположен между двумя домами и задается минимальным номером дома, между которыми он находится. Так переход задающийся числом x проходит между домами с номерами x и $x + 1$.
- Вдоль проспекта на правой его стороне. Переход также расположен между двумя домами и задается аналогично.
- Переход поперек проспекта. Позволяет перейти с одной стороны на другую, и расположен между двумя противоположными домами. Задается номером этих домов.

Желе начинает свой путь у дома с номером ноль, находящегося с левой стороны проспекта. Желе может двигаться как в порядке увеличения номеров домов, так и в порядке уменьшения. Офис находится на правой стороне в доме с номером f .

Желе ненавидит переходы, поэтому хочет минимизировать количество использований переходов на своем пути.

Формат входного файла

В первой строке входного файла дано три натуральных числа n, m, k ($1 \leq n, m, k \leq 10^5$) — количество переходов первого, второго и третьего типа соответственно.

Во второй строке дано n чисел a_i ($0 \leq a_i \leq 10^5$) — числа задающие переходы на левой стороне проспекта.

В третьей строке дано m чисел b_i ($0 \leq b_i \leq 10^5$) — числа задающие переходы на правой стороне проспекта.

В четвертой строке дано k чисел c_i ($0 \leq c_i \leq 10^5$) — числа задающие переходы поперек проспекта.

В пятой строке дано число f ($0 \leq f \leq 10^5$) — номер дома, в который направляется Желе.

Формат выходного файла

В единственной строке выходного файла выведите наименьшее количество раз, которым Желе придется воспользоваться переходами при движении до офиса.

Пример

| streets.in | streets.out |
|--|-------------|
| 1 1 1 2 3 2 1 | 1 |
| 4 3 3 1 4 5 8 8 9 11 0 7 15 14 | 4 |

Задача I. Правописание

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

Франкенштейну очень нравится Марта, но он не знает, как ей об этом сказать. Все обдумав и взвесив, он решил написать ей письмо, в котором признается ей в любви. Вот только с правописанием у него большие проблемы, ведь он все-таки большой и неуклюжий.

И вот Франкенштейн набросал в голове текст письма и сел за работу. Он решил, что текст получится наиболее красивым, если за все время его написания отрывать ручку от бумаги как можно меньше раз. Но и лишний раз проводить по одному и тому же месту — сложно. Под «одним местом» имеется в виду линия ненулевой длины. Также он решил, что каждую букву текста будет писать отдельно, чтобы точно не запутаться. За основу написания букв Франкенштейн взял картинку из найденного им учебника по английскому языку, вот она:



Таким образом, например, чтобы написать букву «А», Франкенштейну надо оторвать ручку от бумаги 2 раза (после того, как он заканчивает писать букву, он тоже отрывает ручку от бумаги), а чтобы написать букву «В» — 1 раз.

Формат входного файла

В первой и единственной строке входного файла содержится текст, который хочет написать Франкенштейн. Текст состоит из заглавных букв латинского алфавита и его длина не превышает 1000.

Формат выходного файла

В единственной строке выходного файла выведите минимальное количество раз, которое Франкенштейну придется оторвать ручку от бумаги, чтобы написать текст.

Примеры

| <code>text.in</code> | <code>text.out</code> |
|----------------------|-----------------------|
| A | 2 |
| WA | 3 |
| AC | 3 |
| IAMGROOT | 13 |