

Вузовско-академическая олимпиада  
по информатике

2021-2022 учебный год

Задания второго  
(заключительного) этапа

## Задача А. Друзья и экзамены

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В мультивселенной есть другая версия Екатеринбурга и ФИИТ УрФУ, где сейчас начинается сессия. Экзамены у них — это целые хакатоны, которые идут по несколько дней подряд **без перерывов**. В этом семестре у студентов  $n$  экзаменов. Также перед студентом есть выбор: сдать  $i$ -й экзамен с  $fl_i$  по  $fr_i$  день или с  $sl_i$  по  $sr_i$ . Понятно, что  $sl_i > fr_i$ , так как преподавателям тоже нужно отдыхать. Люди в параллельном Екатеринбурге очень порядочные, поэтому и экзамены у них идут по порядку, то есть:  $sr_i < fl_{i+1}$ .

Сессия сессией, но у студента ФИИТ Семёна много запланированных с друзьями поездок, а именно  $m$  штук. Поездка номер  $i$  проходит с  $l_i$  по  $r_i$  день включительно. Так как компания друзей одна и та же, поездки по дням не пересекаются.

Естественно Семён не пропустит ни один экзамен, иначе его отчислят. То есть в выбранные промежутки для экзаменов он не может отдыхать с друзьями. Но он может пойти на один из двух вариантов экзамена, а во время другого уехать на отдых. Вот он и задался вопросом, сколько максимум раз он съездит с друзьями отдохнуть при оптимальном выборе дат для сдачи всех экзаменов.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $m$  — количество экзаменов и количество запланированных с друзьями поездок ( $1 \leq n, m \leq 10^5$ ).

В следующих  $n$  строках через пробел заданы целые числа  $fl_i, fr_i, sl_i$  и  $sr_i$  — два промежутка, в один из которых Семён может сдать  $i$ -й экзамен ( $1 \leq fl_i \leq fr_i < sl_i \leq sr_i \leq 10^9, sr_i < fl_{i+1}$  при  $i \leq n - 1$ ).

В следующих  $m$  строках через пробел заданы целые числа  $l_i$  и  $r_i$  — дни, на которые запланирована  $i$ -я поездка с друзьями ( $1 \leq l_i \leq r_i \leq 10^9, r_i < l_{i+1}$  при  $i \leq m - 1$ ).

### Формат выходных данных

Выведите единственное число — максимальное количество раз, которое Семён может съездит с друзьями отдохнуть при оптимальном выборе дат для сдачи **всех** экзаменов.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	24	$l_i = r_i$ для всех $i$	
2	18	$n \leq 10^3, m \leq 10$	
3	22	$n, m \leq 10^3$	2
4	36	$n, m \leq 10^5$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примере из условия**.

**Пример**

стандартный ввод	стандартный вывод
3 5 3 5 7 9 43 50 51 58 100 103 111 114 6 6 47 47 101 101 103 103 112 112	4

## Задача В. Как пройти на кухню

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

### Это интерактивная задача.

Офис компании NAUMEN состоит из  $N$  комнат, соединённых переходами. В этом офисе недавно появилась кухня. Конечно же, на кухне есть печенье, и недавно устроившемуся на работу Владу не терпится их попробовать. Только дороги на кухню он не знает.

Назовём *маршрутом* последовательность комнат без повторений, соединённых переходами. Естественно, любые две комнаты соединены маршрутом. Чтобы максимизировать площадь полезных помещений, офис NAUMEN обходится минимальным возможным количеством переходов. Нетрудно видеть, что для этого достаточно  $N - 1$  переходов, и при этом любые две комнаты соединены ровно одним маршрутом.

Во время перерыва Влад шатается по офису в попытках найти кухню. Он может зайти в любую комнату и спросить, в какой стороне кухня, на что ему вежливо покажут направление. Но перерыв не резиновый, и Влад может задать такой вопрос не больше 20 раз. После чего ему хватит времени только на то, чтобы добежать до кухни и схватить одну печенючку. Помогите Владу найти кухню за приемлемое время!

### Протокол взаимодействия

Взаимодействие начинается с чтения структуры офиса.

В первой строке дано единственное число  $N$  — количество комнат в офисе ( $2 \leq N \leq 2 \cdot 10^5$ ).

В каждой из следующих  $(N - 1)$  строк через пробел даны два числа  $u, v$  — номера комнат, между которыми есть переход ( $1 \leq u, v \leq N$ ). Комнаты пронумерованы от 1 до  $N$ .

Далее Вы можете сделать не более 20 запросов. Чтобы сделать запрос, выведите в отдельной строке:

- $c$  — номер комнаты, в которой вы задаёте вопрос ( $1 \leq c \leq N$ ). В ответ на это Вам сообщат  $b$  — номер соседней комнаты, которая ближе всего к кухне ( $1 \leq b \leq N$ ), либо 0, если Влад уже находится на кухне.

Ответ, как и запросы, выведите в отдельной строке. Вывод ответа не считается запросом при подсчёте их количества. Чтобы сообщить ответ, выведите:

- $! x$  — номер комнаты, являющейся кухней ( $1 \leq x \leq N$ ).

После этого ваша программа должна завершить работу.

После вывода очередного запроса обязательно используйте функции очистки потока, чтобы часть вашего вывода не осталась в каком-нибудь буфере. Например, на C++ надо использовать функцию `fflush(stdout)`, на Java вызов `System.out.flush()`, на Pascal `flush(output)` и `stdout.flush()` для языка Python.

Обратите внимание, что интерактор не является адаптивным. То есть, расположение кухни фиксировано и не изменяется во время работы программы.

## Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	12	$N \leq 20$	
2	17	$N \leq 80$ , существует маршрут, проходящий через все комнаты	
3	23	$N \leq 10^5$ , существует маршрут, проходящий через все комнаты	2
4	20	$N \leq 10^5$ , любой маршрут имеет не более 40 переходов	1
5	28	$N \leq 2 \cdot 10^5$	1, 2, 3, 4

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примере из условия**.

## Пример

стандартный ввод	стандартный вывод
3	
1 3	
3 2	
1	3
0	1
	! 1

## Задача С. Натуральные числа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вадим работает в научно-исследовательском отделении Яндекса. Программировать ему приходится нечасто, а вот изучать самое простое понятие в математике — натуральные числа — приходится каждый день. Как только он не соединял между собой натуральные числа! Вот и сегодня он решил соединить их по-своему.

Два различных числа  $a, b$  соединены, если либо  $a$  — наибольший не равный  $b$  делитель  $b$ , либо  $b$  — наибольший не равный  $a$  делитель  $a$ .

Назовём *путём* последовательность различных натуральных чисел  $[a_1, a_2, \dots, a_k]$  такую, что для всех  $i$  из  $[1, k - 1]$   $a_i$  и  $a_{i+1}$  соединены и  $k \geq 1$ .

Назовём *ценой* пути минимальное значение  $a_i$  в последовательности.

Теперь Вадиму стало интересно про некоторые пары чисел, существует ли между ними путь и, если он есть, то какой наибольшей цены он может быть.

### Формат входных данных

В первой строке дано единственное число  $N$  — количество пар чисел, про которые хочет узнать Вадим ( $1 \leq N \leq 5 \cdot 10^5$ ).

В каждой из следующих  $N$  строк через пробел даны по два числа  $a_i, b_i$  — очередная пара чисел, про которую хочет узнать Вадим ( $1 \leq a_i, b_i \leq 10^7$ ).

### Формат выходных данных

Выведите  $N$  строк. В  $i$ -й строке выведите  $-1$ , если пути между  $a_i$  и  $b_i$  нет, иначе выведите единственное целое число — наибольшую цену пути между  $a_i$  и  $b_i$ .

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	14	$N \leq 10, a_i, b_i \leq 10$	
2	15	$N \leq 10^3, a_i, b_i \leq 10^3$	1
3	21	$N \leq 10^3, a_i, b_i \leq 10^5$	1, 2
4	19	$N \leq 10^5, a_i, b_i \leq 10^5$	1, 2, 3
5	31	$N \leq 5 \cdot 10^5, a_i, b_i \leq 10^7$	1, 2, 3, 4

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примере из условия**.

### Пример

стандартный ввод	стандартный вывод
3	4
4 4	1
4 5	1
4 6	

### Замечание

Для того, чтобы ускорить ввод на C++, используйте `ios_base::sync_with_stdio(false)` и `cin.tie(NULL)`.

Для того, чтобы ускорить работу программы на Python, ускорьте ввод, используя `sys.stdin.readlines()`, а также сдавайте решения под компилятором PyPy 7.3.3.

## Задача D. Руны в поле

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Однажды Вадим гулял по полю и представлял его в виде плоскости с декартовыми координатами. Тут он увидел бесконечное количество рун, которые лежали в точках с координатами  $(x, y)$  такими, что  $x, y$  — натуральные взаимно простые числа, то есть  $\text{НОД}(x, y) = 1$ . Не нужно объяснять, что их цена на чёрном рынке очень высока, поэтому Вадим решил использовать все свои силы, чтобы собрать их.

С собой у него была сеть, которая представляет из себя выпуклый многоугольник. Вадим кинул её на землю, после чего начал собирать попавшие в неё руны. Руна попадает в сеть, если она оказывается либо внутри, либо на границе выкинутой сети. Времени думать у Вадима нет, поэтому он попросил Вас помочь ему посчитать количество рун в сети.

### Формат входных данных

В первой строке дано единственное число  $N$  — количество вершин многоугольника, описывающего кинутую сеть ( $3 \leq N \leq 2 \cdot 10^5$ ).

В каждой из следующих  $N$  строк через пробел даны по два числа  $x_i, y_i$  — координаты вершин этого многоугольника в порядке обхода против часовой стрелки ( $1 \leq x_i, y_i \leq 2 \cdot 10^6$ ). Координаты даны ровно с тремя знаками после запятой.

### Формат выходных данных

Выведите количество рун, попавших в сеть.

Заметьте, что несмотря на вещественные координаты данного многоугольника, от Вас требуется вывести точное значение.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	8	$N \leq 5, x_i, y_i \leq 20$	
2	20	$N \leq 50, x_i, y_i \leq 10^3$	1
3	15	$N \leq 100, x_i, y_i \leq 10^4$	1, 2
4	27*	$N \leq 10^5, x_i, y_i \leq 10^6$	1, 2, 3
5	30	$N \leq 2 \cdot 10^5, x_i, y_i \leq 2 \cdot 10^6$	1, 2, 3, 4

Баллы за все подзадачи начисляются, если решение **корректно работает на примерах из условия**.

Баллы за 1-ю, 2-ю, 3-ю и 5-ю подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

В 4-й подзадаче 9 тестов, каждый тест независимо оценивается в 3 балла. Баллы за эту подзадачу начисляются только если все тесты необходимых подзадач успешно пройдены.

### Пример

стандартный ввод	стандартный вывод
3 1.000 1.000 3.000 1.000 1.000 3.000	5

## Задача Е. Шпион

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вадим — известный шпион. Начальство приказало выкрасть сверхсекретную фотографию из офиса одной сверхсекретной компании.

Сверхсекретную фотографию хранят в одном помещении в  $n$  копиях, каждая копия записана электронно на  $m$  пронумерованных от 0 до  $m - 1$  дискетах. Дискеты с одинаковым номером друг от друга неотличимы, то есть чтобы получить одну копию сверхсекретной фотографии, надо собрать набор из любых дискет с номерами  $0, 1, \dots, m - 1$ .

Вадим проник в это помещение и обнаружил, что эти копии лежат друг за другом на  $nm$  полках, причем на  $i$ -й полке дискета имеет номер  $(i - 1) \bmod m$ , где операция  $a \bmod b$  обозначает взятие остатка от целочисленного деления  $a$  на  $b$ .

Вадим услышал сигнал тревоги — его рассекретили! Поэтому было принято решение сбежать, захватив с собой дискеты с  $q$  непересекающихся отрезков подряд идущих полок:  $[l_1, r_1], [l_2, r_2], \dots, [l_q, r_q]$ .

Вадиму удалось сбежать, но сколько копий он смог вынести? Помогите ему посчитать это количество!

### Формат входных данных

В первой строке заданы целые числа  $n, m$  и  $q$  — количество копий, дискет в каждой из копий и количество отрезков полок, с которых Вадим смог взять дискеты ( $1 \leq n, m \leq 10^9, 1 \leq q \leq 10^5$ ).

В следующих  $q$  строках вводятся целые числа  $l_i, r_i$  — непересекающиеся отрезки, на которых Вадим взял дискеты ( $1 \leq l_i \leq r_i \leq nm$ ).

### Формат выходных данных

Выведите единственное целое число — количество копий, которое взял Вадим.

### Система оценки

Подзадача	Баллы	Ограничения	Необх. подз.
1	26	$n \leq 10^5, m = 2, q \leq 10^5$	
2	35	$n \cdot m \leq 2 \cdot 10^5, q \leq 10^5$	1
3	26	$n \leq 2 \cdot 10^5, m \leq 2 \cdot 10^5, q \leq 10^5$	1, 2
4	13	$n \leq 10^9, m \leq 10^9, q \leq 10^5$	1, 2, 3

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

### Пример

стандартный ввод	стандартный вывод
3 5 4 1 4 15 15 9 12 6 7	1

## Задача F. Катансонны

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ваним придумал классную настольную игру — Катансонны. Игра происходит на клетчатом поле  $W \times H$ . В каждой клетке может находиться либо дом, либо ферма, либо ничего.

Цель игры — вырастить как можно больше тыквачков. Для этого нужно отправлять жителей работать на фермы. Всего на поле  $n$  домов и  $m$  ферм. Каждый дом имеет координаты  $(hx_i, hy_i)$ , и в нём изначально живут  $a_i$  людей. Каждая ферма имеет координаты  $(fx_i, fy_i)$ , и на ней есть  $b_i$  рабочих мест.

Чтобы перемещаться между домами и фермами, жителям нужны кони. Будем считать, что у каждого жителя есть конь, и у всех коней есть одинаковая лошадиная сила  $D$ . С точным значением лошадиной силы Ваним ещё не определился, это число будет определено позже в процессе балансировки игры. Житель может переместиться из дома в ферму или из фермы в дом, если Манхэттенское расстояние между ними не превосходит  $D$ . То есть, между  $i$ -м домом и  $j$ -й фермой можно передвигаться в обе стороны, если  $|hx_i - fx_j| + |hy_i - fy_j| \leq D$ . Из дома в дом, а также из фермы в ферму перемещаться запрещено по правилам игры. Вместимость домов и ферм неограничена: в любой момент в каждом доме и в каждой ферме может находиться сколько угодно людей.

В конце игры ведётся подсчёт тыквачков. На каждой ферме вырастает столько тыквачков, сколько рабочих мест на ней занято. Иными словами, на ферме вырастает один тыквачок за каждого находящегося там человека, но не больше, чем рабочих мест на этой ферме. Дома очки не приносят.

Ваним хочет, конечно же, собрать максимальный счёт. Но он не ищет лёгких путей, поэтому он хочет сделать это при наименьшей возможной лошадиной силе. Помогите ему и скажите, какое максимальное количество тыквачков достижимо в игре, и какая минимальная лошадиная сила позволит набрать это количество.

### Формат входных данных

В первой строке записано два целых числа  $W$  и  $H$  — ширина и высота поля в клетках ( $1 \leq W, H \leq 10^9$ ).

Во второй строке записано два целых числа  $n$  и  $m$  — количество домов и ферм соответственно ( $1 \leq n, m \leq 10^4$ ).

В следующих  $n$  строках через пробел записано по три целых числа  $hx_i, hy_i$  и  $a_i$  — координаты очередного дома и начальное количество людей в нём ( $1 \leq hx_i \leq W, 1 \leq hy_i \leq H, 1 \leq a_i \leq 10^9$ ).

В следующих  $m$  строках через пробел записано по три целых числа  $fx_i, fy_i$  и  $b_i$  — координаты очередной фермы и количество рабочих мест в ней ( $1 \leq fx_i \leq W, 1 \leq fy_i \leq H, 1 \leq b_i \leq 10^9$ ).

Гарантируется, что в каждой клетке не находится два строения одновременно.

### Формат выходных данных

В единственной строке выведите через пробел два целых числа — максимальный возможный счёт в игре, и минимальную возможную лошадиную силу для достижения максимального счёта.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	19	$n = 1, m = 1$	
2	23	$H = 1, n \leq 10^4, m \leq 10^4, hx_i < fx_j$ для всех $i$ и $j$	
3	11	$n \leq 500, m \leq 500$	1
4	27	$H = 1, n \leq 10^4, m \leq 10^4$	1, 2
5	20*	$n \leq 10^4, m \leq 10^4$	1, 2, 3, 4

Баллы за каждую подзадачу **кроме последней** начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примере из условия**.

В последней подзадаче 20 тестов, каждый оценивается независимо и стоит 1 балл. Гарантируется, что из этих 20 тестов хотя бы 10 сгенерированы случайно. Генерация происходит следующим способом: сначала вручную задаются параметры  $W, H, n$  и  $m$ , затем случайно выбирается  $n$  домов и  $m$  ферм на поле, и в конце случайно присваиваются параметры  $a_i$  и  $b_i$ . Баллы за последнюю подзадачу начисляются только если все тесты необходимых подзадач успешно пройдены, а также решение **корректно работает на примере из условия**.

Обратите внимание на необычные ограничения во второй подзадаче. В ней гарантируется, что все дома находятся левее ферм.

### Пример

стандартный ввод	стандартный вывод
11 11 4 4 1 1 2 7 1 1 1 11 2 11 11 1 4 4 1 10 4 2 4 11 1 8 11 2	6 7

## Задача G. Валентин и bash 3

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Пять лет назад Валентин начал изучение командной строки GNU/Linux. Он быстро освоил командную строку, текстовый редактор Vim и язык bash. Но время взяло своё, и Валентин уже не помнит большую часть фиш командной строки.

Сегодня Валентин нашёл свой старый ноутбук тех времён. Решив поностальгировать, он открыл папку «Фотографии», и обнаружил там  $N$  файлов. Название каждого файла являлось строкой из  $K$  символов. Валентин захотел найти там самые запомнившиеся ему моменты, но обнаружил, что в командной строке включен *нечёткий поиск*. Он заключается в том, что когда в консоль вводится запрос из  $K$  букв, ищутся все файлы, название которых отличается от запроса не более чем в одной позиции. Если таких файлов больше одного, вылезит надоедливая подсказка.

Валентин уже не помнит, как отключить нечёткий поиск. Помогите ему для каждого файла определить, какой запрос ввести, чтобы этот файл был единственным подходящим по правилам нечёткого поиска. Посчитайте также количество таких запросов.

Для названий файлов используется алфавит из  $A$  символов. Все эти символы являются символами ASCII с кодами от 33 до 126 включительно. Подходящими ответами будут считаться только запросы, использующие этот алфавит.

### Формат входных данных

В первой строке через пробел дано три целых числа  $N$ ,  $K$  и  $A$  — количество файлов, длина названия файла, и размер алфавита ( $1 \leq N \leq 10\,000$ ,  $1 \leq K \leq 16$ ,  $1 \leq A \leq 94$ ).

Во второй строке дано  $A$  различных символов с кодами от 33 до 126 включительно — алфавит, использующийся для названий файлов и ответов.

Далее идут  $N$  строк. В каждой строке дана последовательность из  $K$  символов — название  $i$ -го файла. Гарантируется, что используются только символы из алфавита, и что нет двух файлов с одинаковыми названиями.

### Формат выходных данных

Выведите  $N$  строк. В  $i$ -й строке выведите целое число — количество запросов, для которых этот файл является единственным подходящим. Если существует хотя бы один подходящий запрос (то есть, число больше нуля), через пробел выведите любой такой запрос.

Подходящими считаются только запросы, использующие символы из алфавита, описанного во входных данных.

### Система оценки

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	15	$N \leq 50$ , $K \leq 4$ , $A \leq 26$	
2	16	$N \leq 200$ , $K \leq 4$ , $A \leq 26$	1
3	16	$N \leq 1000$ , $K \leq 4$ , $A \leq 94$	1, 2
4	34	$N \leq 1000$ , $K \leq 16$ , $A \leq 94$	1, 2, 3
5	19	$N \leq 10\,000$ , $K \leq 16$ , $A \leq 94$	1, 2, 3, 4

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены, а также решение **корректно работает на примерах из условия**.

## Примеры

стандартный ввод	стандартный вывод
5 4 8 abcdenst nest test tent tens abcd	20 ness 13 tesn 13 ttnt 20 tets 29 abcd
3 2 3 abc aa ab ba	0 1 cb 1 bc