

# Общая информация по задачам олимпиады

## Ограничение на размер исходного кода программы

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

## Ограничение на посылку решений

По каждой задаче на проверку принимается не более 50 решений.

По каждой задаче участник не может отправить решение более одного раза в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

## Процесс тестирования

Обратите внимание, что у каждой подзадачи есть список необходимых подзадач. Подзадача будет тестироваться, если все тесты во всех ее необходимых подзадачах пройдены. Будьте внимательны, некоторые подзадачи могут не тестироваться, если не пройдены примеры, подходящие под ограничения этих подзадач.

## Система оценки

На нашем туре есть два типа оценки подзадач: «тест» и «подзадача». «Тест» означает, что в подзадаче баллы за каждый тест выставляются независимо, и сумма баллов за все тесты равна числу баллов за всю подзадачу. «Подзадача» означает, что баллы начисляются только за полностью пройденную подзадачу.

Для более подробной информации по оценке подзадач читайте пункт «Система оценки» в каждой задаче.

За каждую задачу выставляется суммарный балл по всем ее подзадачам. В каждой подзадаче оценивается лучшее решение, то есть за подзадачу выставляется максимальный набранный по ней балл среди всех решений.

## Получение информации о результатах проверки

Чтобы получить информацию о проверке вашего решения используйте ссылку «Информация о проверке» во вкладке «Решения» в PCMS2 Web Client.

Для более подробной информации по оценке подзадач читайте пункт «Получение информации о результатах проверки» в каждой задаче.

## Задача А. File names

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Петя и Вася любят писать свои операционные системы, поэтому на компьютере Пети установлена операционная система PetOS, а на компьютере Васи — VasOS.

Однажды Петя захотел передать Васе несколько файлов со своего компьютера, но выяснилось, что сделать это не так просто. Дело в том, что в PetOS именем файла может быть любая строка, состоящая из латинских букв и точек, длиной от 1 до 20 символов, а в VasOS поддерживаются только имена файлов вида `<filename>.<extension>`, где `<filename>` и `<extension>` — непустые строки, состоящие из латинских букв, при этом длина `<filename>` должна быть не больше 8, а длина `<extension>` — не больше 3.

Помогите Пете по списку имен файлов понять, сколько из них он сможет передать на компьютер Васи, не меняя их имя.

### Формат входных данных

Первая строка содержит число  $n$  — число файлов в списке Пети ( $1 \leq n \leq 100$ ). Следующие  $n$  строк содержат имена файлов. Все имена имеют длину от 1 до 20 символов и содержат только строчные латинские буквы и точки.

### Формат выходных данных

Выведите одно число — число файлов, которые Петя может передать Васе без переименования.

### Система оценки

В этой задаче все тесты оцениваются независимо.

### Получение информации о результатах проверки

Сообщается результат проверки и количество заработанных баллов на каждом тесте.

### Пример

стандартный ввод	стандартный вывод
8 .exe script a.in.txt a.t con.exe solution.cpp b.java verylongname.txt	3

### Пояснение к примеру

В приведенном примере можно передать файлы: `a.t`, `con.exe`, `solution.cpp`.

## Задача В. Neatness

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Мальчики Дима и Митя живут в общежитии в одной комнате и убираются в ней через каждые  $k$  дней по очереди. Наступил новый месяц из  $n$  дней, и ребятам нужно составить новое расписание уборки. Чтобы это сделать, ребята просто выбирают, кто из них будет прибираться в этом месяце первым, и в какой день из первых  $k$  дней месяца это произойдет. В следующий раз уборка будет назначена ровно через  $k$  дней, и убираться должен будет второй мальчик. Например, если Дима по расписанию убирается в день номер  $i$ , то в день  $i + k$  по расписанию будет убираться Митя, в день  $i + 2k$  снова Дима и так далее.

Расписание должно быть честным, то есть оба мальчика должны прибрать комнату одинаковое число раз. Всё осложняется тем обстоятельством, что и Дима, и Митя планируют в этом месяце совершить по одной поездке на олимпиады. Если один из мальчиков отсутствует в свою очередь, за него убирается другой, при этом расписание уборки никак не сдвигается. Известно, что Дима будет отсутствовать в дни с  $a$  по  $b$  включительно, а Митя — с  $c$  по  $d$  включительно. Дни месяца нумеруются целыми числами от 1 до  $n$ . Поездки на олимпиады не пересекаются, то есть в любой день месяца хотя бы один мальчик будет дома.

Помогите ребятам определить, кого нужно поставить в расписании первым и в какой день (из первых  $k$ ), или скажите, что составить подходящее расписание невозможно.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $k$  — количество дней в месяце и периодичность уборки ( $2 \leq n \leq 10^{18}$ ,  $1 \leq k \leq n$ ).

Во второй строке записаны целые числа  $a$ ,  $b$ ,  $c$  и  $d$ , описывающие поездки мальчиков на олимпиады ( $1 \leq a \leq b \leq n$ ;  $1 \leq c \leq d \leq n$ ;  $b < c$  или  $d < a$ ).

### Формат выходных данных

Если расписание составить невозможно, выведите  $-1$ .

Иначе в первой строке выведите натуральное число — номер дня, с которого ребятам нужно начать расписание, во второй строке выведите «Dima», если первым должен убираться Дима, и «Mitya» иначе.

Если подходящих вариантов несколько, выведите любой из них.

### Система оценки

Подзадача	Баллы	Ограничения	Оценка	Необх. подзадачи
1	17	$n \leq 100$ ; $k \leq 100$	подзадача	—
2	24	$n \leq 10^5$ ; $k \leq 10^5$	подзадача	1
3	33	$n \leq 10^{18}$ ; $k \leq 10^5$	подзадача	1, 2
4	26	$n \leq 10^{18}$ ; $k \leq 10^{18}$	подзадача	1, 2, 3

### Получение информации о результатах проверки

В подзадачах 1, 2 и 3 сообщается результат проверки на каждом тесте.

В подзадаче 4 сообщаются баллы, а также результат проверки на первом непройденном тесте.

## Примеры

стандартный ввод	стандартный вывод
12 5 6 10 11 12	3 Dima
18 4 15 18 1 6	3 Mitya
10 3 1 4 5 6	-1

## Пояснения к примерам

Представим тесты из примеров в виде таблиц со следующими обозначениями:

- «d» — день, когда отсутствует Дима,
- «m» — день, когда отсутствует Митя,
- «D» — день, когда будет убираться Дима,
- «M» — день, когда будет убираться Митя,
- «\*» — день, с которого можно начать расписание.

Тогда для первого примера получим следующее:

Дни	1	2	3	4	5	6	7	8	9	10	11	12
Отъезды	*	*	*	*	*	d	d	d	d	d	m	m
По графику	.	.	D	.	.	.	.	M	.	.	.	.
На самом деле	.	.	D	.	.	.	.	M	.	.	.	.

Так, каждый будет прибираться по одному разу, и отъезды график уборки не изменят.  
Таблица для второго примера:

Дни	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Отъезды	m*	m*	m*	m*	m	m	.	.	.	.	.	.	.	.	d	d	d	d
По графику	.	.	M	.	.	.	D	.	.	.	M	.	.	.	D	.	.	.
На самом деле	.	.	D	.	.	.	D	.	.	.	M	.	.	.	M	.	.	.

Из-за отъезда Мити в третий день уборку будет делать Дима, но Митя будет дежурить 15-го числа из-за отсутствия Димы. Так, каждый будет прибираться по два раза, хоть и не всегда по графику.

## Задача C. Tax Collection

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Байтландия представляет собой прямоугольник  $n \times m$ , разделенный на  $n \cdot m$  квадратных провинций. Недавно в Байтландии провели налоговую реформу, в результате которой для каждой провинции была зафиксировано число  $a[i, j]$ . Если  $a[i, j] > 0$ , значит провинция, находящаяся в квадрате  $(i, j)$  должна каждый месяц платить в бюджет  $a[i, j]$  байткоинов. Если же  $a[i, j] < 0$ , то провинция  $(i, j)$  является дотационной и получает из бюджета  $-a[i, j]$  байткоинов.

Для сбора налогов правительство разработало следующую схему. В одной из провинций будет построено здание казначейства. Каждый месяц из этого здания будет выезжать сборщик налогов. Он будет объезжать все провинции, собирая налоги и выдавая дотации, и возвращаться обратно. Его путь должен удовлетворять следующим свойствам:

- путь должен начинаться в провинции, в которой находится казначейство,
- путь должен заканчиваться в провинции, имеющей общую сторону с провинцией, в которой находится казначейство,
- каждая провинция должна быть посещена **ровно** один раз,
- соседние провинции в пути должны быть иметь общую сторону.

Правительство хочет выбрать провинцию для казначейства и путь для сборщика таким образом, чтобы для каждой дотационной провинции сборщик мог выдать им нужную сумму денег из уже собранных ранее. Помогите им построить такой путь или скажите, что это невозможно.

### Формат входных данных

В первой строке заданы два целых числа  $n$  и  $m$  — размеры Байтландии ( $2 \leq n, m \leq 300$ ).

Далее следует  $n$  строк, каждая из которых состоит из  $m$  целых чисел  $a[i, j]$ . Данные строки описывают провинции:  $a[i, j]$  — значение, на которое изменится количество байткоинов у сборщика при посещении провинции, находящейся в позиции  $(i, j)$  ( $1 \leq |a[i, j]| \leq 10\,000$ ).

### Формат выходных данных

Если существует решение, то выведите  $n \cdot m$  пар чисел — координаты провинций, которые должен посетить сборщик, в том порядке, в котором он должен их посетить. В случае, если решения не существует, выведите  $-1$ .

### Система оценки

Подзадача	Баллы	Ограничения			Оценка	Необх. подзадачи
		$n$	$m$	Доп. ограничения		
1	23	$n = 2$	$m \leq 300$	—	подзадача	—
2	15	$n \leq 300$	$m \leq 300$	все $a_{i,j} > 0$	подзадача	—
3	29	$n \leq 300$	$m \leq 300$	ровно одно $a_{i,j} < 0$	подзадача	—
4	33	$n \leq 300$	$m \leq 300$	—	подзадача	1, 2, 3

### Получение информации о результатах проверки

Сообщается результат проверки на каждом тесте каждой подзадачи.

## Примеры

стандартный ввод	стандартный вывод
2 3 -3 4 2 1 -5 3	1 2 1 3 2 3 2 2 2 1 1 1
4 4 1 -5 -3 1 1 5 2 -2 4 1 -3 1 -8 6 -2 3	2 3 2 2 1 2 1 1 2 1 3 1 4 1 4 2 3 2 3 3 4 3 4 4 3 4 2 4 1 4 1 3
2 2 1 -2 -1 1	-1
3 3 1 1 1 1 1 1 1 1 1	-1

## Замечание

Путь для первого примера:

-3	4	2
1	-5	3

Сумма, которая есть у сборщика после каждой провинции: 4, 6, 9, 4, 5, 2.

Путь для второго примера:

1	-5	-3	1
1	5	2	-2
4	1	-3	1
-8	6	-2	3

Сумма, которая есть у сборщика после каждой провинции: 2, 7, 2, 3, 4, 8, 0, 6, 7, 4, 2, 5, 6, 4, 5, 2.

## Задача D. Road Building

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Близится чемпионат мира по поеданию бургеров. Чтобы попробовать бургеры разных производителей, было решено провести данное мероприятие в двух городах. Но организаторы столкнулись с проблемой: нет прямой дороги между городами. В кратчайшие сроки было решено построить её, однако между городами находится лес, в котором растет  $n$  деревьев. Так как времени почти не осталось, решили не пилить деревья, а построить прямую дорогу максимальной ширины, чтобы она не пересекала никакое дерево. А чтобы путешествие между городами было более живописным, решили оставить по обе стороны хотя бы по одному дереву.

Для удобства представим деревья как окружности на декартовой плоскости. Вам необходимо построить дорогу бесконечной длины, параллельно прямой, которая задается двумя точками, чтобы по обе стороны было хотя бы одно дерево и ширина была максимальна.

### Формат входных данных

В первой строке заданы четыре целых числа  $x_s, y_s, x_f$  и  $y_f$  — координаты точек  $(x_s, y_s)$  и  $(x_f, y_f)$ , через которые проходит прямая.

Во второй строке записано число  $n$  — количество деревьев в лесу ( $2 \leq n \leq 2 \cdot 10^5$ ). Никакие два дерева не пересекаются, но могут касаться друг друга.

В следующих  $n$  строках содержится описание каждого дерева. В  $i$ -й из этих строк задано три целых числа  $x_i, y_i$  и  $r_i$  — координаты центра дерева и радиус ( $1 \leq r_i \leq 10^9$ ).

Все координаты не превосходят  $10^9$  по абсолютному значению.

### Формат выходных данных

Необходимо вывести максимальную ширину дороги, которая параллельна прямой, проходящей через точки  $(x_s, y_s)$  и  $(x_f, y_f)$ . По обе стороны дороги должно быть хотя бы одно дерево. Дорога может касаться деревьев, но не пересекать их. Если построить такую дорогу нельзя, то выведите 0.

Ваш ответ будет засчитан, если его абсолютная или относительная ошибка не превосходит  $10^{-6}$ . Формально, пусть ваш ответ равен  $a$ , а ответ жюри равен  $b$ . Ваш ответ будет засчитан, если  $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ .

### Система оценки

Подзадача	Баллы	Ограничения	Оценка	Необх. подзадачи
1	27	$2 \leq n \leq 100$	подзадача	—
2	19	Радиусы одинаковые	подзадача	—
3	23	Прямая параллельна осям координат	подзадача	—
4	31	Нет дополнительных ограничений	подзадача	1, 2, 3

### Получение информации о результатах проверки

В подзадачах 1, 2 и 3 сообщается результат проверки на каждом тесте.

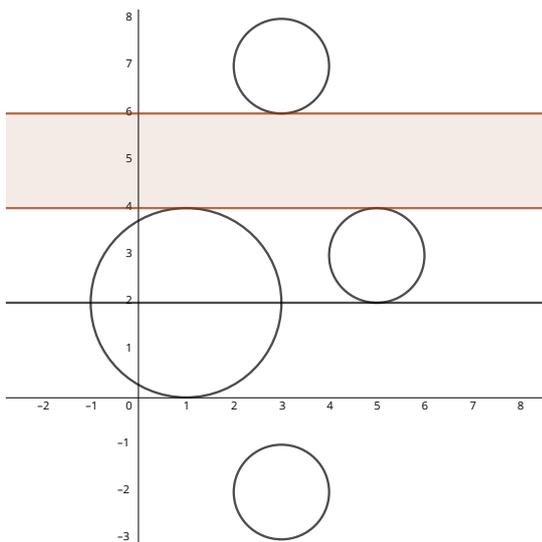
В подзадаче 4 сообщаются баллы, а также результат проверки на первом непройденном тесте.

## Примеры

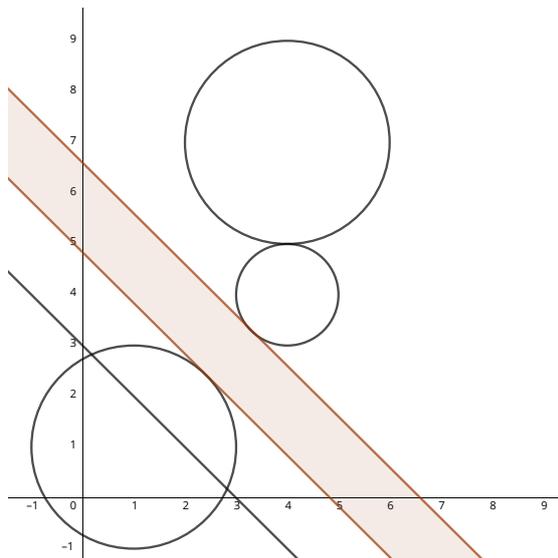
стандартный ввод	стандартный вывод
0 2 4 2 4 5 3 1 3 -2 1 1 2 2 3 7 1	2.0000000000000000
0 3 3 0 3 1 1 2 4 4 1 4 7 2	1.242640687119285

## Пояснения к примерам

Первый пример:



Второй пример:



## Задача E. The Expert

Ограничение по времени: 5 секунд  
Ограничение по памяти: 512 мегабайт

Нарисовать семь прямых красных линий, все они должны быть перпендикулярны друг другу, причем некоторые должны быть нарисованы зеленым цветом, некоторые — прозрачным, плюс одна — в форме котенка.

Техническое задание

Вам поручили нарисовать на плоскости  $n$  прямых, параллельных осям координат. Никакие две прямые не должны совпадать. Про некоторые пары прямых известно, что они должны быть параллельны, а про некоторые пары — что они должны быть перпендикулярны. Все прямые должны задаваться уравнениями вида  $a \cdot x + b \cdot y + c = 0$ , где  $a$ ,  $b$  и  $c$  — целые числа. Пусть  $i$ -я прямая задается уравнением  $a_i \cdot x + b_i \cdot y + c_i = 0$ . От вас, помимо удовлетворения всех условий на перпендикулярность и параллельность, требуется минимизировать размер множества, содержащего все  $a_i$ ,  $b_i$  и  $c_i$ . Другими словами, требуется минимизировать количество различных чисел среди всех коэффициентов.

Определите минимальное количество чисел, которое необходимо использовать, чтобы выполнить задание, или сообщите, что задание выполнить невозможно. Если задание можно выполнить, найдите любой способ это сделать, используя минимальное количество различных чисел.

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $m$  — количество прямых и количество условий ( $1 \leq n, m \leq 10^6$ ).

В следующих  $m$  строках содержится описание условий на перпендикулярность или параллельность пар прямых. В  $i$ -й строке содержатся три целых числа  $t_i$ ,  $p_i$  и  $q_i$ , если  $t_i$  равно 0, прямые с номерами  $p_i$  и  $q_i$  должны быть параллельны, иначе, прямые с номерами  $p_i$  и  $q_i$  должны быть перпендикулярны ( $t_i \in \{0, 1\}$ ;  $1 \leq p_i, q_i \leq n$ ;  $p_i \neq q_i$ ).

### Формат выходных данных

Если удовлетворить все условия невозможно, выведите  $-1$ .

Если задание выполнимо, выведите минимальное количество чисел, необходимое для выполнения задания. В следующих  $n$  строках выведите целые числа  $a_i$ ,  $b_i$  и  $c_i$  — коэффициенты  $i$ -й прямой. Коэффициенты не должны по модулю превышать  $10^9$ .

### Система оценки

Подзадача	Баллы	Ограничения	Оценка	Необх. подзадачи
1	10	$n, m \leq 15$	подзадача	—
2	20	$n \leq 5\,000$ , все $t_i = 0$	подзадача	—
3	20	$m = n - 1$ , $p_i = 1$ , $q_i = i + 1$	подзадача	—
4	20	$n \leq 5\,000$	подзадача	1, 2
5	15	$n \leq 100\,000$	подзадача	1, 2, 4
6	15	Нет дополнительных ограничений	подзадача	1, 2, 3, 4, 5

### Получение информации о результатах проверки

Во всех подзадачах сообщаются баллы, а также результат проверки на первом непройденном тесте.

## Примеры

стандартный ввод	стандартный вывод
3 3 1 3 2 0 1 3 1 1 2	2 -7 0 0 0 -7 -7 -7 0 -7
2 2 1 1 2 0 2 1	-1

## Пояснения к примерам

В первом примере можно использовать, например, следующие прямые:

- $-7 \cdot x + 0 \cdot y + 0 = 0$
- $0 \cdot x - 7 \cdot y - 7 = 0$
- $-7 \cdot x + 0 \cdot y - 7 = 0$

Во втором примере прямые 1 и 2 должны быть как перпендикулярны, так и параллельны. Такие прямые построить нельзя.