

# Общая информация по задачам первого тура

## Доступ к результатам проверки решений задач во время тура

Во всех задачах вы можете неограниченное число раз запрашивать результат окончательной проверки. Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.

## Ограничение на размер исходного кода программы-решения

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

## Процесс тестирования

Во всех задачах баллы за подзадачу начисляются только при прохождении всех тестов подзадачи. В таблице системы оценивания для каждой подзадачи указаны номера *необходимых подзадач*. Тесты для данной подзадачи запускаются только, если все необходимые подзадачи пройдены.

## Ограничения

Задача	Ограничение по времени	Ограничение по памяти
1. Оборона крепости	2 секунды	512 МБ
2. Экспериментальная робототехника	2 секунды	512 МБ
3. Ловить или не ловить	1 секунда	512 МБ
4. Обитаемые горы	2 секунды	512 МБ

## Замечания

В задачах с большим размером входных данных жюри рекомендует участникам, использующим языки группы C/C++, отправлять решения на проверку с использованием компилятора LINUX GNU C++ и считывать данные, используя `scanf`.

## Задача 1. Оборона крепости

Имя входного файла: `castle.in`  
Имя выходного файла: `castle.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Стена осаждённой крепости состоит из  $n$  участков, пронумерованных от 1 до  $n$ . Разведка доложила, что противник планирует отправить на штурм участка стены с номером  $i$  отряд из  $a_i$  нападающих. Для обороны крепости на участки стены будут направлены в общей сложности  $s$  защитников.

Участки стены различаются качеством укрепления, что приводит к различной эффективности обороны: на участке стены с номером  $i$  каждый защитник способен отразить атаку  $k_i$  нападающих.

Пусть на участок с номером  $i$  отправлено  $x_i$  защитников. Тогда если количество нападающих не превышает величину  $x_i \cdot k_i$ , то на этом участке ни один из нападающих не прорвётся в крепость. Иначе в крепость прорвутся  $(a_i - x_i \cdot k_i)$  нападающих.

Требуется написать программу, распределяющую защитников по участкам стены так, чтобы их общее количество было равно  $s$  и в крепость прорвалось наименьшее количество нападающих.

### Формат входных данных

Первая строка входных данных содержит целые числа  $n$  — количество участков стены и  $s$  — количество защитников крепости ( $1 \leq n \leq 100\,000$ ;  $1 \leq s \leq 10^9$ ).

Следующие  $n$  строк содержат по два целых числа  $a_i$ ,  $k_i$  — общее количество нападающих на  $i$ -й участок стены и количество нападающих, атаку которых может отразить один защитник этого участка ( $1 \leq a_i, k_i \leq 10^9$ ).

### Формат выходных данных

Выходные данные должны содержать единственное целое число — минимальное количество нападающих, которые прорвутся в крепость.

### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения				Необх. подзадачи
		$n$	$s$	$a$	$k$	
1	17	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$k_i = 1$	
2	21	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$1 \leq k_i \leq 2$	1
3	23	$1 \leq n \leq 100$	$1 \leq s \leq 10\,000$	$1 \leq a_i \leq 100$	$1 \leq k_i \leq 100$	1, 2
4	39	$1 \leq n \leq 100\,000$	$1 \leq s \leq 10^9$	$1 \leq a_i \leq 10^9$	$1 \leq k_i \leq 10^9$	1 – 3

### Примеры

<code>castle.in</code>	<code>castle.out</code>
1 10 8 1	0
3 3 4 2 1 1 10 8	3

### Пояснения к примерам

В первом тесте ни один из нападающих не прорвется в крепость, если поставить всех 10 защитников на единственный участок, так как они смогут отбить всех нападающих. Во втором примере можно, например, направить двух защитников на первый участок и одного — на третий.

## Задача 2. Экспериментальная робототехника

Имя входного файла:	robots.in
Имя выходного файла:	robots.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Университеты Татарстана являются ведущими центрами по образовательной робототехнике. Для популяризации этого направления школьникам было предложено провести эксперимент по выживанию роботов в ограниченном пространстве.

Эксперимент проходит на прямоугольном поле размером  $n \times m$  клеток. В начале эксперимента в заданных клетках размещается по одному неактивированному роботу. По команде «Старт» запускается таймер, который подаёт сигнал в начале каждой секунды. После каждого сигнала таймера, но не позднее чем через  $T_{max} = 10^9$  секунд после команды «Старт», разрешается активировать некоторых роботов.

Каждая клетка поля закрашена в один из четырёх цветов, распознаваемых сенсором робота. Цвет обозначает направление движения из данной клетки в соседнюю: на север, юг, восток или запад. В момент сигнала таймера каждый активированный робот перемещается в соседнюю клетку в направлении, соответствующем цвету клетки, в которой он находится. Все активированные роботы перемещаются одновременно. Цвета клеток поля выбраны так, что при перемещении никакой робот не может выйти за пределы поля.

Во избежание повреждений запрещается активировать роботов таким образом, что это приведёт к их столкновению. Столкновением называется ситуация, когда два или более активированных роботов оказываются в одной клетке поля. Если происходит столкновение, то эксперимент считается неуспешным. При этом перемещение роботов из соседних клеток навстречу друг другу, в результате которого они меняются местами, к столкновению не приводит.

Эксперимент считается завершённым успешно, если все активированные роботы могут продолжать движение без столкновений по полю сколь угодно долго. Результатом эксперимента является количество активированных роботов.

Требуется написать программу, которая поможет школьникам по описанию поля и клеток, где исходно располагаются неактивированные роботы, определить максимально возможный результат эксперимента и, если потребуется, каких именно роботов и в какие моменты времени следует активировать для достижения такого результата.

### Формат входных данных

В первой строке входных данных записаны целые числа  $n$ ,  $m$  и  $g$ , где  $n$  и  $m$  — размеры поля с севера на юг и с запада на восток соответственно ( $1 \leq n, m \leq 1000$ ), а  $g$  — признак, равный 1 или 0, обозначающий необходимость определения клеток и моментов времени для активации роботов.

Последующие  $n$  строк по  $m$  символов в каждой описывают цвета клеток поля и разрешение на активацию робота в них. Цвет поля задаётся английской буквой, соответствующей направлению движения: «N» или «n» — север, «S» или «s» — юг, «E» или «e» — восток и «W» или «w» — запад.

Клетки, в которых исходно размещены неактивированные роботы, обозначены заглавными буквами, а клетки, в которых исходно робота нет — строчными. Гарантируется, что на поле находится хотя бы один неактивированный робот.

### Формат выходных данных

В первой строке выходных данных выведите одно число  $k$  — максимально возможный результат эксперимента. Для входных данных, в которых значение  $g$  равно 1, в каждой из последующих  $k$  строк выведите три целых числа  $r$ ,  $c$  и  $t$ : номера строки и столбца, описывающие клетку для активации робота, и момент времени для его активации соответственно ( $1 \leq r \leq n$ ;  $1 \leq c \leq m$ ;  $1 \leq t \leq 10^9$ ). Строки нумеруются от 1 до  $n$  сверху вниз (с севера на юг), а столбцы — от 1 до  $m$  слева направо (с запада на восток).

Если стратегий активации роботов, приводящих к максимальному результату несколько, то выведите любую из них.

## Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения			Необх. подзадачи
		$n, m$	$g$	Дополнительно	
1	11	$1 \leq n, m \leq 10$	$g = 0$	в каждой клетке исходно находится робот	
2	13	$1 \leq n, m \leq 100$	$g = 0$	в каждой клетке исходно находится робот	1
3	13	$1 \leq n, m \leq 100$	$g = 0$	—	1, 2
4	23	$1 \leq n, m \leq 100$	$g = 1$	—	1–3
5	17	$1 \leq n, m \leq 1000$	$g = 0$	—	1–3
6	23	$1 \leq n, m \leq 1000$	$g = 1$	—	1–5

## Примеры

robots.in	robots.out
3 4 0 SSSS EESW ENWW	4
3 4 1 SSSS EeSW ENwW	4 2 3 2 3 2 1 3 4 1 2 1 2
4 4 1 essS Eess Snww EeWN	5 1 4 9 2 1 4 4 3 3 4 1 2 4 4 7

## Пояснение к примеру

В первом примере можно активировать любых четырёх роботов, выбрав для этого подходящие моменты времени. Например, роботы, находящиеся в клетках (1, 1) и (3, 1) не могут быть активированы одновременно. Указывать, какие именно роботы должны быть активированы и в какие моменты в времени, в данном тесте не требуется.

Во втором примере приведённый ответ не является единственным.

В третьем примере активированные в клетках (4, 1) и (4, 3) роботы могут перемещаться между клетками (4, 2) и (4, 3) сколько угодно раз, не сталкиваясь при этом.

### Задача 3. Ловить или не ловить

Имя входного файла: fisher.in  
Имя выходного файла: fisher.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Владельцы рыболовецкого судна, ведущего промысел на реке Кама, решили в летнем сезоне оптимизировать свой бизнес.

Они получили сезонное разрешение на лов рыбы в  $n$  точках русла реки на расстояниях  $x_1, x_2, \dots, x_n$  километров от устья. При этом в точке с номером  $i$  разрешается выловить не более  $a_i$  тонн рыбы.

Выловленную рыбу можно продавать на  $m$  оптовых базах, расположенных вдоль берега реки в точках на расстояниях  $y_1, y_2, \dots, y_m$  километров от устья. При этом база в точке номер  $j$  готова в этом сезоне закупить не более  $b_j$  тонн рыбы по цене  $c_j$  рублей за тонну.

Расстояния от устья до точек вылова и оптовых баз измеряются вдоль русла реки.

Судно отправляется на лов из устья реки и должно вернуться туда же после окончания сезона. В течение сезона судно может произвольным образом плавать вверх и вниз по реке, останавливаясь для лова или продажи рыбы. Грузоподъёмность судна достаточна для перевозки любого количества выловленной рыбы. При удалении от устья судно движется против течения, расходуя на один километр пути топливо стоимостью  $p$  рублей. При перемещении в сторону устья судно движется по течению и поэтому не расходует топлива.

По итогам сезона прибыль за улов будет равна суммарной стоимости проданной рыбы за вычетом суммарной стоимости затраченного топлива.

Требуется написать программу, определяющую максимальную прибыль, которую можно получить за сезон.

#### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $m$  и  $p$  — количество точек лова рыбы, количество оптовых баз и стоимость топлива ( $1 \leq n, m \leq 500\,000$ ;  $0 \leq p \leq 10^9$ ).

Следующие  $n$  строк содержат по два целых числа  $x_i$  и  $a_i$  — расстояние от устья и максимальный улов для каждой точки лова рыбы ( $0 < x_1 < x_2 < \dots < x_n \leq 10^9$ ;  $0 < a_i \leq 10^6$ ).

Следующие  $m$  строк содержат по три целых числа  $y_j$ ,  $b_j$ ,  $c_j$  — расстояние от устья, максимальное закупаемое количество тонн рыбы и цена закупки за тонну для каждой оптовой базы ( $0 < y_1 < y_2 < \dots < y_m \leq 10^9$ ;  $0 < b_j, c_j \leq 10^6$ ).

#### Формат выходных данных

Выходные данные должны содержать единственное целое число — максимальную возможную прибыль.

#### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения		Необх. подзадачи
		$n, m$	Дополнительные	
1	16	$1 \leq n, m \leq 50\,000$	$p = 0$	
2	9	$1 \leq n, m \leq 50\,000$	$y_m < x_1$	1
3	16	$1 \leq n, m \leq 50\,000$	$x_n < y_1$	1
4	11	$1 \leq n, m \leq 1\,000$	—	
5	9	$1 \leq n, m \leq 8\,500$	—	4
6	20	$1 \leq n, m \leq 50\,000$	—	1–5
7	6	$1 \leq n, m \leq 200\,000$	—	1–6
8	7	$1 \leq n, m \leq 320\,000$	—	1–7
9	6	$1 \leq n, m \leq 500\,000$	—	1–8

## Примеры

fisher.in	fisher.out
3 2 0 1 5 2 3 4 5 2 2 10 3 6 5	50
2 1 100 6 5 100 4 5 100 2000	9400
3 3 10 1 1 10 100 20 10 2 1000 1 11 50 50 17 50 2	2441

## Пояснения к примерам

Во втором примере оптимальными будут следующие действия. Следует доплыть до точки на расстоянии 6 километров от устья, потратив 600 рублей на топливо, и выловить в ней 5 тонн рыбы. После этого следует спуститься на 1 километр по реке к базе на расстоянии 5 километров от устья и продать выловленную рыбу по цене 2000 рублей за тонну. Затем следует вернуться в устье реки. Суммарная прибыль составит 9400 рублей.

## Задача 4. Обитаемые горы

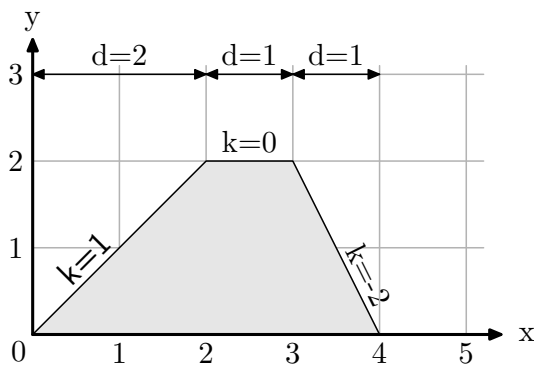
Имя входного файла:	mountain.in
Имя выходного файла:	mountain.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

По мотивам фантастической повести  
А. и Б. Стругацких «Обитаемый остров»

Правительство страны Неизвестных Отцов планирует построить в гористом районе на границе с Хонтией башню противобаллистической защиты.

Участок горной цепи в этом районе представлен ломаной, состоящей из  $n$  звеньев, последовательно соединяющих  $n + 1$  вершину. Вершины пронумерованы числами от 0 до  $n$  в порядке возрастания координаты  $x$ . Звенья пронумерованы числами от 1 до  $n$ , при этом  $i$ -е звено соединяет вершины с номерами  $i - 1$  и  $i$ .

Вершина с номером 0 находится в точке  $(0, 0)$ . Звено с номером  $i$  задано двумя числами  $d_i$  — длиной проекции на ось  $x$  и  $k_i$  — угловым коэффициентом. Таким образом, если вершина с номером  $i - 1$  имеет координаты  $(x_{i-1}, y_{i-1})$ , то координаты  $i$ -й вершины можно вычислить как  $(x_{i-1} + d_i, y_{i-1} + k_i \cdot d_i)$ . Последняя вершина лежит на оси  $x$ , то есть  $y_n = 0$ .



Точка  $A(x_A, y_A)$  находится в *прямой видимости* из точки  $B(x_B, y_B)$ , если ни одна точка отрезка  $AB$  не находится **строго** под ломаной.

Башня представлена вертикальным отрезком ненулевой длины, нижняя точка которого расположена на ломаной. Гражданин страны Неизвестных Отцов чувствует себя в безопасности, если верхняя точка башни находится в его прямой видимости.

Пусть верхняя точка башни имеет координаты  $(x, y)$ . Два разведчика выбегают из нижней точки башни соответственно на запад (в направлении уменьшения координаты  $x$ ) и на восток (в направлении увеличения координаты  $x$ ). Каждый разведчик бежит по поверхности горной цепи до тех пор, пока дальнейшее перемещение не выводит верхнюю точку башни из его прямой видимости или до границы горной цепи.

Правительство подготовило  $q$  вариантов расположения башни, каждый из которых характеризуется двумя целыми числами  $(u_j, v_j)$  — координатами верхней точки башни. Требуется написать программу, которая для каждого варианта определяет  $x$ -координаты двух точек, до которых добегут разведчики.

### Формат входных данных

Первая строка входных данных содержит два числа  $n$  и  $q$  ( $1 \leq n, q \leq 400\,000$ ) — количество звеньев ломаной и количество вариантов расположения башни.

Ограничения на величины последующих входных данных зависят от значения константы  $C$ , которая может быть равна  $10^4$  или  $10^9$  в зависимости от подзадачи (см. таблицу системы оценивания).

Каждая из последующих  $n$  строк содержит по два целых числа  $d_i, k_i$  ( $1 \leq d_i \leq C$ ;  $-C \leq k_i \leq C$ ) — проекцию на ось  $x$  и угловой коэффициент  $i$ -го звена ломаной ( $0 = x_0 < x_1 < \dots < x_i < \dots < x_n \leq C$ ;  $y_0 = y_n = 0$ ;  $-C \leq y_i \leq C$ ).

Каждая из последующих  $q$  строк содержит по два целых числа  $u_j, v_j$  ( $0 \leq u_j \leq C$ ,  $-C \leq v_j \leq C$ ) — координаты верхней точки башни в  $j$ -м варианте расположения.

### Формат выходных данных

Выходные данные должны содержать  $q$  строк, в каждой из которых находятся по два целых числа  $l_j$  и  $r_j$  —  $x$ -координаты двух точек, до которых добегут разведчики, направляющиеся на запад и на восток соответственно в  $j$ -м варианте расположения башни. Гарантируется, что числа  $l_j$  и  $r_j$  являются целыми.

### Примеры

mountain.in	mountain.out
6 1 3 1 2 -1 1 1 1 -1 1 1 2 -1 5 3	3 8
5 3 1 1 1 -2 2 0 2 1 1 -1 3 0 3 5 3 3	1 6 0 7 0 6
6 4 1 2 2 -2 1 1 1 -2 4 1 1 -1 1 4 3 4 10 4 7 4	0 4 1 9 4 10 1 10
8 4 1 -3 2 0 1 1 2 0 1 -3 1 3 1 2 1 0 2 -2 6 -1 6 4 7 -4	0 6 4 9 0 10 6 9



### Таблица системы оценивания

Номер подзадачи	Баллы	Ограничения			Необх. подзадачи
		$n, q$	$C$	Дополнительные ограничения	
1	9	$1 \leq n, q \leq 100$	$C = 10^4$	$k_i = \pm 1$	
2	9	$1 \leq n, q \leq 100$	$C = 10^4$		1
3	10	$1 \leq n, q \leq 3000$	$C = 10^9$		1, 2
4	11	$1 \leq n, q \leq 100\,000$	$C = 10^9$	$k_i = \pm 1$	1
5	11	$1 \leq n, q \leq 100\,000$	$C = 10^9$	нижние точки всех башен совпадают	
6	12	$1 \leq n, q \leq 100\,000$	$C = 10^9$	верхние точки всех башен находятся на одной горизонтальной прямой	
7	21	$1 \leq n, q \leq 100\,000$	$C = 10^9$		1–6
8	17	$1 \leq n, q \leq 400\,000$	$C = 10^9$		1–7

### Замечание

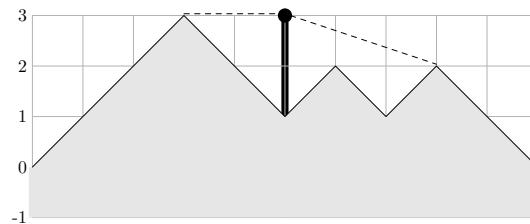


Рис. 1: Первый пример

Обратите внимание, что все точки отрезка между (6, 2) и (7, 1) находятся в прямой видимости из верхней точки башни согласно определению в условии.

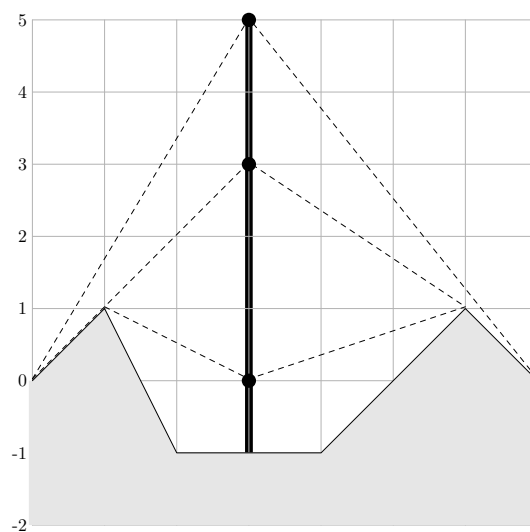


Рис. 2: Второй пример

В данном тесте нижние точки всех вариантов башен совпадают.

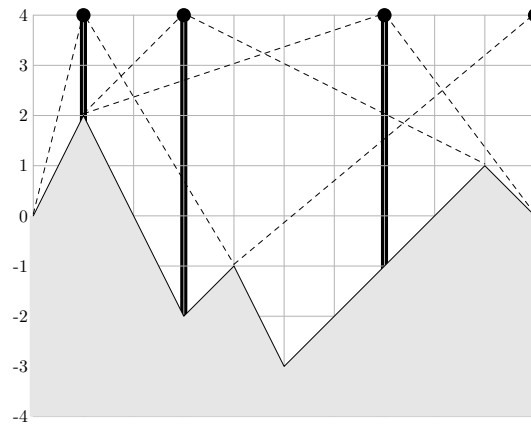


Рис. 3: Третий пример

В данном тесте верхние точки башен всех вариантов находятся на одной горизонтальной прямой. Обратите внимание, что нижняя точка башни может совпадать с одним из концов горной цепи.

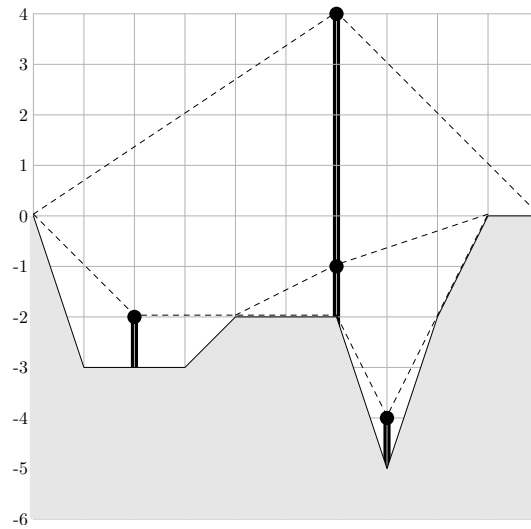


Рис. 4: Четвертый пример

В четвёртом тесте показывается, что в стране Неизвестных Отцов горная цепь может целиком находиться ниже уровня её концов.