



# Общая информация по задачам олимпиады

## Ограничение по памяти

Во всех задачах ограничение составляет 512 МБ.

## 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 двух раз в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

## Система оценки

Каждая задача олимпиады поделена на несколько подзадач. Чтобы набрать баллы по подзадаче, программа должна пройти все тесты этой подзадачи.

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. По каждой задаче вам будет доступна информация по количеству набранных баллов в каждой подзадаче или результат проверки на первом непройденном тесте.

## Таблица результатов

Во время соревнования доступна текущая таблица результатов. Для доступа к ней используйте ссылку «Результаты» в PCMS2 Web Client. Таблица результатов в PCMS2 Web Client не является окончательной. **В последние 30 минут соревнования таблица результатов не будет показываться.**



## Задача А. Parking Problem

Ограничение по времени: 2 секунды

Паулина приехала на конференцию по фрагментации файловых систем в университет Иннополис на автомобиле. Парковка университета представляет из себя полосу ширины  $n$  метров, на которой параллельно друг другу стоят транспортные средства. На ней могут парковаться владельцы автомобилей и мотоциклов. Кроме того, вся парковка разделена на  $n$  зон ширины 1 метр. Водители уважают друг друга и паркуются по разделителям: мотоцикл занимает одну зону ширины 1 метр, а автомобиль — две соседние зоны суммарной ширины 2 метра.

Паулина хочет въехать на парковку. В данный момент, некоторые зоны заняты транспортными средствами. Также, у парковки собралась очередь на въезд из  $m$  транспортных средств, не считая автомобиля Паулины. Все транспортные средства по очереди будут въезжать на парковку, и занимать любое свободное место: одну зону, если это мотоцикл, и две соседние зоны, если это автомобиль. Если водитель не может найти ни одного свободного места для своего транспортного средства, то он покидает Иннополис, не занимая места на парковке.

Паулине скоро выступать на конференции, и так как в Иннополисе люди очень вежливые, все в очереди готовы пропустить Паулину на парковку перед собой, чтобы она успела на свой доклад. Паулина не хочет пользоваться своим положением, но опасается, что если мест на парковке не хватит, то на доклад она точно не попадет. Так как Паулина эксперт во фрагментации файловых систем, она понимает, что транспортные средства не всегда удачно занимают места на парковке, и ей может не остаться места. Она смогла быстро понять, заехав перед кем в очереди, она гарантирует своему автомобилю свободное место на парковке.

Вам задано, какие из  $n$  зон на парковке заняты, а также очередь из  $m$  мотоциклов и автомобилей, не считая автомобиля Паулины. Требуется понять для каждого  $i$  от 0 до  $m$ , останется ли место для автомобиля Паулины, если она въедет на парковку после первых  $i$  транспортных средств из очереди, вне зависимости от того, какие зоны они займут.

### Формат входных данных

Вам нужно будет решить несколько тестов.

В первой строке задано целое число  $t$  — число тестов во входных данных ( $1 \leq t \leq 50\,000$ ).

Далее заданы  $t$  тестов, каждый из которых состоит из двух строк.

В первой из этих строк записано  $n$  символов ‘.’ и ‘X’,  $i$ -й из которых описывает  $i$ -ю зону: ‘.’ для свободной зоны и ‘X’ — для занятой ( $1 \leq n \leq 10^5$ ).

Во второй из этих строк записано  $m$  заглавных букв английского алфавита ‘C’ и ‘M’, описывающих очередь из автомобилей ‘C’ и мотоциклов ‘M’ ( $1 \leq m \leq n$ ). Первая буква соответствует началу очереди, а последняя — концу.

Сумма  $n$  по  $t$  заданным тестам не превосходит  $5 \cdot 10^5$ .

### Формат выходных данных

Выведите  $t$  строк: по одному для каждого теста.

Строка должна содержать  $m + 1$  заглавных букв английского алфавита ‘Y’ и ‘N’, для каждого  $i$  от 0 до  $m$  запишите ‘Y’, если Паулина найдет место для своего автомобиля, въехав на парковку после  $i$  первых транспортных средств из очереди, вне зависимости от того, какие зоны они займут, и ‘N’ в противном случае.

### Система оценки

Подзадача	Баллы	Ограничения
1	31	на парковке нет занятых зон
2	22	на парковке не более одной занятой зоны
3	23	в очереди только мотоциклы
4	24	нет дополнительных ограничений



## Примеры

стандартный ввод	стандартный вывод
1 X..X MM	YNN
5 ..... MMMC ..... CCCM .X..X. MMM XXXXXX CMMCM ..... MM	YYYN YYNN YNN NNNNN YYY

## Замечание

В первом примере Паулина может поставить свой автомобиль на две свободные зоны, если заедет первой. Если же Паулина заедет не первой, то свободных зон будет одна или ни одной, тогда она не сможет припарковаться.

В первом тесте второго примера после того, как заедут три первых мотоцикла в очереди, они могут встать следующим образом: «М.М.М.», тогда Паулина не сможет найти две соседние зоны, чтобы припарковать свой автомобиль.



## Задача В. Diamond Hands

Ограничение по времени: 2 секунды

У компании «Бриллиантовые руки» длинная и неоднозначная история. Начиная со дня основания, у нее было много и удачных, и неудачных дней. Для простоты будем считать, что в удачный день цена акции компании увеличивалась на единицу, а в неудачный день — уменьшалась на единицу. Как это часто бывает, удачные для компании дни шли длинными подряд идущими отрезками. Впрочем, то же самое можно сказать и про неудачные дни. Третьего не дано: каждый день был либо удачным, либо неудачным.

Вы хотите понять, какие отрезки дней были для компании удачными, а какие неудачными. Чтобы это сделать, вы добыли исторические данные цен акций компании в виде  $n$  пар  $(d_i, p_i)$ , означающие, что через  $d_i$  дней после выпуска акций разница с изначальной ценой составляла  $p_i$  единиц ( $p_i$  может быть произвольным целым числом, в том числе любым отрицательным).

Постройте описание истории компании в виде **минимального** числа отрезков удачных или неудачных дней, либо определите, что в данных ошибка, и восстановить подходящую под данные историю невозможно. Если существует несколько ответов с минимальным числом отрезков, выведите любой из них.

### Формат входных данных

Первая строка содержит число  $n$  ( $1 \leq n \leq 200\,000$ ). Следующие  $n$  строк содержат пары чисел  $d_i p_i$  ( $1 \leq d_i \leq 10^8$ ;  $-10^8 \leq p_i \leq 10^8$ ;  $d_i < d_{i+1}$  для всех  $i$  от 1 до  $n - 1$ ).

### Формат выходных данных

Если в исторические данные закралась ошибка, выведите  $-1$ . Иначе, в первой строке выведите  $k$  — количество отрезков дней. В каждой из следующих  $k$  строк выведите пару  $l_i c_i$  ( $1 \leq l_i \leq 10^8$ ;  $c_i \in \{+, -\}$ ), означающую, что очередной отрезок длился  $l_i$  дней и состоял из удачных дней, если  $c_i = +$ , либо из неудачных дней, если  $c_i = -$ .

Описание отрезков дней должно идти в их хронологическом порядке, начиная со дня выпуска акций и заканчивая в день  $d_n$ , то есть, сумма всех  $l_i$  должна быть равна  $d_n$ .

### Система оценки

Подзадача	Баллы	Ограничения
1	6	$n \leq 2000$ , если ответ существует, то $k = 1$
2	8	$n \leq 2000$ , если ответ существует, то $k \leq 2$
3	10	$n \leq 200\,000$ , если ответ существует, то $k \leq 2$
4	28	$n \leq 2000$ , $d_i, p_i \leq 2000$
5	17	$n \leq 2000$ , $d_i, p_i \leq 10^8$
6	31	$n \leq 200\,000$ , $d_i, p_i \leq 10^8$

### Примеры

стандартный ввод	стандартный вывод
4 2 2 3 3 5 1 7 1	3 3 + 3 - 1 +
2 3 -3 7 -3	2 5 - 2 +
1 1 0	-1



## Замечание

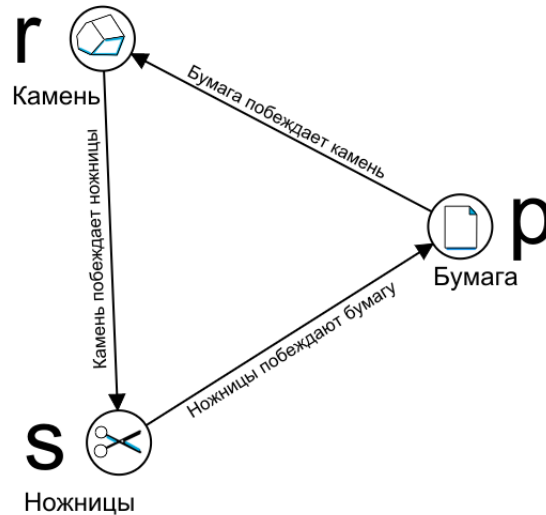
В первом примере, первые три дня были удачными, а значит, через 2 дня после выпуска акций разница составляла 2, а через 3 дня после выпуска разница составляла 3. За тремя удачными днями следовали три неудачных, и после 5 дней разница с изначальной ценой стала равна 1, а через 6 дней цена сравнялась с исходной. Последний, седьмой день был успешным, и финальная разница, после 7 дней, равна 1.



## Задача C. RPS string

Ограничение по времени: 2 секунды

В Иннополисе новое соревнование: битва роботов в камень-ножницы-бумага. Правда все крутые роботы заняты полноценной работой, поэтому в боях участвуют самые простые боты. Каждый из них всегда выбрасывает что-то одно: камень, ножницы или бумагу.



Роботы стоят в линию. Дальше судья может выбрать два соседних робота, и заставить их сыграть один раунд в камень-ножницы-бумага. Если один из них выиграл другого, то проигравший выбывает из игры и его убирают из списка роботов, оставшиеся роботы при этом не меняют свой порядок в линии. При этом есть две версии правил. В первой из них, если роботы показали одно и то же, судья может самостоятельно выбрать, какого из них удалить из игры. Во второй же версии, если роботы показали одно и то же, оба робота остаются. Робот побеждает, если все остальные роботы выбыли из игры.

Однако судье стало очень скучно вести бои, поэтому он хочет для каждого робота узнать, может ли он так выбирать пары в каждом раунде, чтобы этот робот победил. Помогите ему решить эту задачу!

### Формат входных данных

Вам нужно будет решить несколько тестов.

В первой строке задано целое число  $t$  ( $t \geq 1$ ) — число тестов во входных данных.

Далее заданы  $t$  тестов, каждый из которых состоит из целого числа и строки.

Число  $d$  обозначает версию правил, которая используется в текущей расстановке роботов  $d \in \{1, 2\}$ .

Дальше идёт строка  $s$ , в которой записано  $n$  ( $1 \leq n \leq 5 \cdot 10^5$ ) строчных букв английского алфавита 'r', 'p' и 's', описывающих список роботов. Точнее символы 'r', 'p' и 's' означают роботов, которые всегда показывают камень, бумагу и ножницы соответственно.

Сумма  $n$  по  $t$  заданным тестам не превосходит  $5 \cdot 10^5$ .

### Формат выходных данных

Выведите  $t$  строк: по одному для каждого теста.

Строка должна содержать  $n$  цифр '0' и '1', для каждого  $i$  от 1 до  $n$  запишите '1', если судья может так выбирать пары для раундов чтобы  $i$ -й робот победил, и '0' в противном случае.

### Система оценки



Подзадача	Баллы	Ограничения
1	7	$d = 1, \sum n \leq 20$
2	9	$d = 1, \sum n \leq 200$
3	10	$d = 1, \sum n \leq 5000$
4	11	$d = 1, \sum n \leq 5 \cdot 10^5$
5	7	$\sum n \leq 20$
6	13	$\sum n \leq 200$
7	10	$\sum n \leq 5000$
8	33	$\sum n \leq 5 \cdot 10^5$

## Пример

стандартный ввод	стандартный вывод
4	10111
1 grspp	001
2 pps	10101
1 rpsps	110010001
2 rpssprsr	

## Замечание

Разберём первый тестовый пример: «grspp»,  $d = 1$ . Пронумеруем роботов от 1 до 5 слева направо.

- Для того чтобы остался робот с номером 1, судья может действовать следующим образом:
  - Судья выбирает роботов с номерами 4 и 5, они выбрасывают одно и то же, поэтому судья решает что 5-й выбывает. Остаются роботы с номерами {1, 2, 3, 4}.
  - Судья выбирает роботов с номерами 2 и 3, 2-й выбывает. Остаются роботы с номерами {1, 3, 4}.
  - Судья выбирает роботов с номерами 3 и 4, 4-й выбывает. Остаются роботы с номерами {1, 3}.
  - Судья выбирает роботов с номерами 1 и 3, 3-й выбывает. Остался робот с номером 1.
- Для робота с номером 2, судья не может избавиться от 1-го робота, не удаляя 2-го, поэтому 2-й не может победить.
- Для того чтобы остался робот с номером 3, судья может действовать следующим образом:
  - Судья выбирает роботов с номерами 4 и 5, они выбрасывают одно и то же, поэтому судья решает что 5-й выбывает. Остаются роботы с номерами {1, 2, 3, 4}.
  - Судья выбирает роботов с номерами 1 и 2, 1-й выбывает. Остаются роботы с номерами {2, 3, 4}.
  - Судья выбирает роботов с номерами 2 и 3, 2-й выбывает. Остаются роботы с номерами {3, 4}.
  - Судья выбирает роботов с номерами 3 и 4, 4-й выбывает. Остался робот с номером 3.
- Для того чтобы остался робот с номером 4, судья может действовать следующим образом:
  - Судья выбирает роботов с номерами 4 и 5, они выбрасывают одно и то же, поэтому судья решает что 5-й выбывает. Остаются роботы с номерами {1, 2, 3, 4}.
  - Судья выбирает роботов с номерами 2 и 3, 2-й выбывает. Остаются роботы с номерами {1, 3, 4}.
  - Судья выбирает роботов с номерами 1 и 3, 3-й выбывает. Остаются роботы с номерами {1, 4}.



- Судья выбирает роботов с номерами 1 и 4, 1-й выбывает. Остался робот с номером 4.
5. Для того чтобы остался робот с номером 5, судья может действовать следующим образом:
- Судья выбирает роботов с номерами 4 и 5, они выбрасывают одно и то же, поэтому судья решает что 4-й выбывает. Остаются роботы с номерами {1, 2, 3, 5}.
  - Судья выбирает роботов с номерами 2 и 3, 2-й выбывает. Остаются роботы с номерами {1, 3, 5}.
  - Судья выбирает роботов с номерами 1 и 3, 3-й выбывает. Остаются роботы с номерами {1, 5}.
  - Судья выбирает роботов с номерами 1 и 5, 1-й выбывает. Остался робот с номером 5.

Разберём второй тестовый пример: «pps»,  $d = 2$ . Пронумеруем роботов от 1 до 3 слева направо.

1. Для робота с номером 1, судья не может избавиться от 2-го робота, не удаляя 1-го, поэтому 1-й не может победить.
2. Для робота с номером 2, судья не может избавиться от 1-го робота, не удаляя 2-го, поэтому 2-й не может победить.
3. Для того чтобы остался робот с номером 3, судья может действовать следующим образом:
  - Судья выбирает роботов с номерами 2 и 3, 2-й выбывает. Остаются роботы с номерами {1, 3}.
  - Судья выбирает роботов с номерами 1 и 3, 1-й выбывает. Остался робот с номером 3.





## Задача D. Long puzzle

Ограничение по времени: 2 секунды

Вы собираете одномерный пазл. Каждая его деталь характеризуется тремя параметрами: длиной, типом левой границы и типом правой границы. Границы бывают трех типов: плоская, выпуклая и вогнутая. Детали нельзя разворачивать, то есть менять местами левую и правую границы. Любая выпуклая граница стыкуется с любой вогнутой и наоборот. Стыковать детали плоскими границами нельзя.

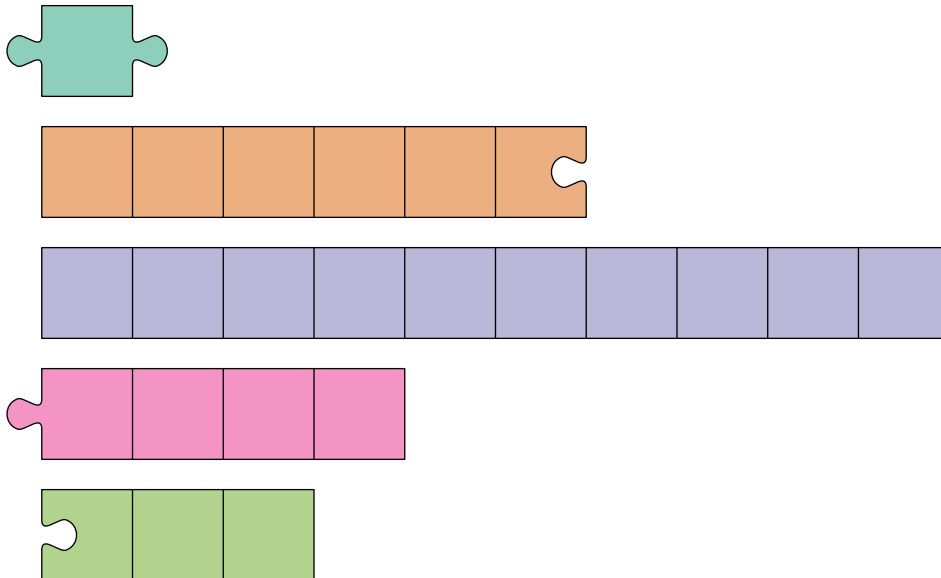


Рис. 1: Примеры деталей

Вы хотите соединить последовательно несколько деталей (возможно, одну), чтобы получился кусок длины  $l$ . Левая и правая границы куска должны быть плоскими. Найдите количество подмножеств деталей, из которых можно собрать требуемый кусок, по модулю 1 000 000 007. Обратите внимание, что вам нужно найти именно количество различных подмножеств деталей, а не количество различных способов, которыми можно детали соединить.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $l$  — количество деталей пазла и требуемая длина собранного куска ( $1 \leq n \leq 300$ ,  $1 \leq l \leq 300$ ).

В следующих  $n$  строках дано описание деталей пазла. Каждая строка содержит  $a_i$ ,  $b_i$  и  $c_i$  — длину детали, тип левой границы и тип правой границы, соответственно ( $1 \leq a_i \leq l$ ;  $b_i, c_i \in \{\text{«in»}, \text{«out»}, \text{«none»}\}$ ). Строка «in» соответствует вогнутой границе, «out» — выпуклой, «none» — плоской.

### Формат выходных данных

Выведите одно целое число — остаток от деления количества подмножеств деталей, из которых можно собрать требуемый кусок, на 1 000 000 007.

### Система оценки

Подзадача	Баллы	Ограничения
1	20	$n \leq 20$
2	20	$b_i \in \{\text{«in»}, \text{«none»}\}, c_i \in \{\text{«out»}, \text{«none»}\}$
3	20	$n, l \leq 50$
4	20	$n, l \leq 100$
5	20	Без дополнительных ограничений



### Примеры

стандартный ввод	стандартный вывод
5 10 1 out out 6 none in 10 none none 4 out none 3 in none	3
4 5 1 none out 1 in out 2 in out 1 in none	1

### Замечание

Детали пазла из первого примера соответствуют иллюстрации выше.

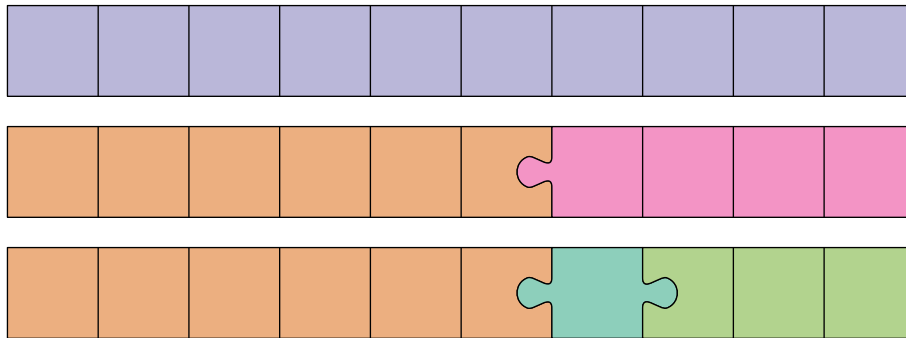


Рис. 2: Подмножества деталей, из которых можно собрать кусок требуемой длины



## Задача E. Rooted MST

Ограничение по времени: 3 секунды

Вам дан простой неориентированный граф содержащий  $n+1$  вершин пронумерованных  $0, 1, \dots, n$  и  $n + m$  ребер.

Вес ребра между вершинами  $0$  и  $i$  равен  $a_i$  для  $1 \leq i \leq n$ .

Вес ребра между вершинами  $u_i$  и  $v_i$  равен  $w_i$  для  $1 \leq i \leq m$ .

Вам нужно ответить на  $q$  запросов, каждый запрос содержит два целых числа  $i, w$ , вам нужно поменять вес ребра между вершинами  $0$  и  $i$  на  $w$  и найти вес минимального остовного дерева в графе.

Обратите внимание, что запросы перманентны, т.е. изменения остаются навсегда.

### Формат входных данных

В первой строке записаны два целых числа  $n, m$  ( $2 \leq n \leq 300\,000, 0 \leq m \leq 300\,000$ ).

Во второй строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

В каждой из следующих  $m$  строк записаны три целых числа  $u_i, v_i, w_i$  ( $1 \leq u_i, v_i \leq n, 0 \leq w_i \leq 10^9$ ).

Гарантируется, что данный граф простой, а именно не содержит петель и кратных ребер.

В следующей строке записано одно целое число  $q$  ( $1 \leq q \leq 300\,000$ ).

В каждой из следующих  $q$  строк записаны два целых числа  $i, w$  ( $1 \leq i \leq n, 1 \leq w \leq 10^9$ ).

### Формат выходных данных

В каждой строке выведите одно целое число — вес минимального остовного дерева в графе после  $i$  запросов.

### Система оценки

Подзадача	Баллы	Ограничения
1	10	$n, m, q \leq 2000$
2	10	Все веса равны 1 или 2
3	10	$w = 1$ во всех запросах
4	10	$i = 1$ во всех запросах
5	10	$i \leq 5$ во всех запросах
6	10	$m = n - 1, u_i = v_i - 1$
7	20	$n, m, q \leq 150\,000$
8	20	Нет дополнительных ограничений



## Пример

стандартный ввод	стандартный вывод
5 7	6
3 2 1 2 1	6
1 5 1	5
1 3 2	5
2 5 2	5
4 5 2	6
3 4 1	6
2 4 2	6
1 2 1	6
10	5
3 2	
2 3	
4 1	
3 2	
5 1	
5 3	
3 1	
2 3	
4 3	
5 1	