



# 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 одного раза 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.



## Задача А. Expression Formatting

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Маша учится программировать, но преподаватель часто ее ругает за то, что ее программы выглядят неаккуратно. Например, Маша очень не любит ставить в выражениях пробелы, и всегда пишет их слитно, например так:  $(a+b)*c$ . Преподаватель же заставляет Машу ставить пробелы вокруг знаков операций, чтобы было так:  $(a + b) * c$ . Маша не хочет тратить время на такие глупости, поэтому просит вас написать программу, которая ставит пробелы за нее.

### Формат входных данных

Первая строка содержит выражение, в котором надо расставить пробелы. Выражение может содержать переменные, знаки арифметических операций: «+», «-», «\*» или «/», и скобки. Длина строки не более 200 символов.

Выражение не содержит пробелов и является корректным. Все переменные состоят из одной строчной буквы английского алфавита.

### Формат выходных данных

Выведите то же выражение, добавив по одному пробелу до и после каждого знака операции.

### Система оценки

В задаче 20 тестов, каждый тест оценивается независимо от других в 5 баллов. Число баллов за решение — сумма баллов за каждый пройденный тест. Вам сообщаются результаты тестирования на всех двадцати тестах.

### Примеры

стандартный ввод	стандартный вывод
$a+b$	$a + b$
$((a))-b+(c*(d))$	$((a)) - b + (c * (d))$
$(a)/(b-b)+((d)+((c)))$	$(a) / (b - b) + ((d) + ((c)))$



## Задача В. Contest Rescheduling

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

В последнее время становится все больше олимпиад по программированию. Чтобы участникам было удобнее, организаторы стараются, чтобы олимпиады не пересекались по времени, однако иногда такое случается.

Однажды две олимпиады были запланированы в один день. Первая олимпиада должна была начаться в момент времени  $s_1$  и длиться  $d_1$  (таким образом, заканчиваясь в момент  $s_1 + d_1$ ), вторая же должна начаться в момент времени  $s_2$  и длиться  $d_2$  (заканчиваясь в  $s_2 + d_2$ ). Длительность олимпиад изменять нельзя, однако начало можно двигать по времени. При этом оба организатора имеют следующие ограничения на передвижение олимпиад. Первая олимпиада должна начаться не раньше  $l_1$  и закончиться не позже  $r_1$ , вторая — начаться не раньше  $l_2$  и закончиться не позже  $r_2$ .

Ваша задача — написать программу, которая бы помогла организаторам олимпиад перенести их таким образом, чтобы они не пересекались, и при этом выполнялись все ограничения. Если решений несколько, вам требуется найти такое, в котором суммарное изменение времени начала олимпиад, минимально. То есть, если олимпиады нужно начать в моменты времени  $c_1$  и  $c_2$ , то вам нужно минимизировать величину  $(|s_1 - c_1| + |s_2 - c_2|)$ .

### Формат входных данных

Входные данные содержат несколько тестов.

Первая строка содержит целое число  $n$  — число тестов, которые нужно решить ( $1 \leq n \leq 50\,000$ )

Далее следуют описания  $n$  тестов. Описание каждого теста состоит из двух строк.

Первая из этих строк содержит четыре целых числа  $l_1$ ,  $r_1$ ,  $l_2$  и  $r_2$  — ограничение на время проведения олимпиад ( $0 \leq l_i < r_i \leq 10^9$ ).

Вторая из этих строк содержит четыре целых числа  $s_1$ ,  $d_1$ ,  $s_2$  и  $d_2$  — времена начала олимпиад и их длительности ( $l_i \leq s_i$ ;  $s_i + d_i \leq r_i$ ).

### Формат выходных данных

Выведите  $n$  строк. В  $i$  строке выведите ответ на  $i$ -й тест.

В качестве ответа на тест выведите два целых числа  $c_1$  и  $c_2$ : времена начала первой и второй олимпиады в новом расписании, соответственно. Если есть несколько решений, минимизирующих суммарный сдвиг, выведите любое. Если невозможно провести олимпиады, удовлетворив все ограничения, выведите два числа  $-1$ .

### Система оценки

Подзадача	Баллы	$n$	$s_i, d_i$	$l_i, r_i$
1	19	$n \leq 1000$	$s_i, d_i \leq 20$	$l_i = 0; r_i = 50$
2	20	$n \leq 1000$	$s_i, d_i \leq 50$	$l_i, r_i \leq 50$
3	21	$n \leq 1000$	$s_i, d_i \leq 10\,000$	$l_i, r_i \leq 10\,000$
4	15	$n \leq 1000$	$s_i, d_i \leq 10^9$	$l_i, r_i \leq 10^9$
5	25	$n \leq 50\,000$	$s_i, d_i \leq 10^9$	$l_i, r_i \leq 10^9$

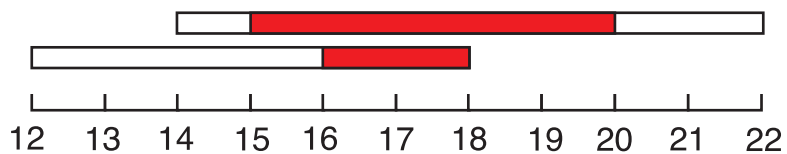
### Пример

стандартный ввод	стандартный вывод
3	16 14
14 22 12 18	-1 -1
15 5 16 2	12 16
12 22 14 20	
14 5 15 4	
12 14 16 18	
12 2 16 2	

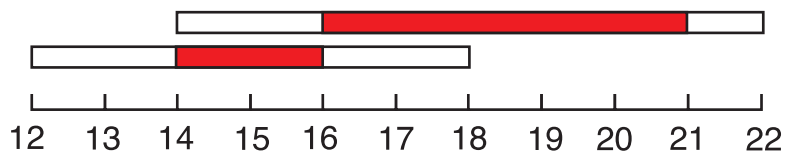


## Пояснение к примеру

Иллюстрация к первому тесту. Расписание олимпиад до сдвига:



Расписание олимпиад после сдвига:



Суммарный сдвиг равен  $|15 - 16| + |16 - 14| = 3$ . Это минимальная возможная сумма.



## Задача C. Advertisement Profit

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Пашка — популярный видеоблогер. Сейчас на его канале ровно 10 000 подписчиков. Он решил составить расписание выпуска роликов на ближайший месяц. Ролики бывают двух типов: обычные и рекламные. У Пашки заготовлено  $n$  обычных и  $m$  рекламных роликов. Выпустить ролик он может в любом порядке. Так как Пашка топовый контентмейкер, после выпуска  $i$ -го обычного ролика, количество подписчиков на его канале увеличится на целое положительное число  $a_i$ . Однако, после выпуска  $i$ -го рекламного ролика, количество подписчиков на его канале уменьшится на целое положительное число  $b_i$ . При этом, если Пашка выпустит  $i$ -й рекламный ролик, и в этот момент на его канале будет  $s$  подписчиков, компания, заказавшая этот ролик, заплатит ему  $s \cdot c_i$  центов. Гарантируется, что даже если Пашка выпустит все запланированные рекламные ролики и только их, количество подписчиков на канале останется неотрицательным.

Пашка еще не определился, сколько роликов выпустить в следующем месяце: у него есть  $q$  различных вариантов. Каждый вариант характеризуется одним целым числом  $d_i$  — количеством роликов, которые он хочет выпустить. Теперь он просит вас для каждого варианта определить, какую максимальную прибыль он может получить в таком случае.

### Формат входных данных

В первой строке дано одно целое число  $n$  — количество заготовленных обычных роликов ( $0 \leq n \leq 100$ ). В следующих  $n$  строках дано по одному целому числу  $a_i$  — изменение количества подписчиков после выпуска  $i$ -го обычного ролика ( $1 \leq a_i \leq 100$ ).

В следующей строке дано одно целое число  $m$  — количество заготовленных рекламных роликов ( $0 \leq m \leq 100$ ). В следующих  $m$  строках дано по два целых числа  $c_i$  и  $b_i$  — количество центов, которые компания заплатит за каждого подписчика на момент выпуска ролика, и изменение количества подписчиков после выпуска  $i$ -го рекламного ролика ( $1 \leq c_i, b_i \leq 100$ ).

В следующей строке дано одно целое число  $q$  — количество различных вариантов, которые хочет проверить Пашка ( $1 \leq q \leq n + m + 1$ ). В следующих  $q$  строках дано по одному целому числу  $d_i$  — количество роликов, которые Пашка хочет выпустить в следующем месяце ( $0 \leq d_i \leq n + m$ ). Гарантируется, что все  $d_i$  — различные.

### Формат выходных данных

Для каждого варианта выведите в новой строке одно число — максимальную прибыль в центах, которую Пашка может получить, если выпустит ровно  $d_i$  роликов.

### Система оценки

Подзадача	Баллы	Ограничения			
		$n$	$m$	$q$	Дополнительно
1	10	$n \leq 5$	$m \leq 5$	$q \leq n + m + 1$	—
2	11	$n \leq 100$	$m \leq 5$	$q \leq n + m + 1$	—
3	16	$n \leq 100$	$m \leq 100$	$q = 1$	$d_1 = n + m$
4	17	$n = 0$	$m \leq 100$	$q \leq n + m + 1$	—
5	19	$n \leq 100$	$m \leq 30$	$q \leq n + m + 1$	$c_i, b_i \leq 10$
6	27	$n \leq 100$	$m \leq 100$	$q \leq n + m + 1$	—



## Примеры

стандартный ввод	стандартный вывод
1 10 2 10 20 20 10 4 0 1 2 3	0 200000 299900 300200
3 10 40 30 3 5 10 15 20 1 100 7 0 1 2 3 4 5 6	0 150000 199900 209870 210710 211340 211550

## Пояснение к примеру

В первом примере, если Пашка выпустит только один ролик, ему нужно выпустить второй рекламный ролик, и тогда он заработает  $10\,000 \cdot 20 = 200\,000$ . Если он хочет выпустить два ролика, тогда сначала надо выпустить второй рекламный ролик, а затем первый рекламный ролик. В таком случае, его прибыль будет равна  $10\,000 \cdot 20 + 9\,990 \cdot 10 = 299\,900$ . Если он хочет выпустить все три ролика, тогда сначала ему нужно выпустить обычный ролик, потом второй рекламный, и затем первый рекламный. Прибыль будет равна  $10\,010 \cdot 20 + 10\,000 \cdot 10 = 300\,200$ .



## Задача D. RSA factoring

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

RSA — криптографическая система, где важную роль играют числа вида  $n = pq$ , где  $p$  и  $q$  — различные простые числа. Число  $n$  называют модулем RSA и используют для дальнейших вычислений. Стойкость RSA основана на том факте, что для известного числа  $n$  не известно достаточно быстрого алгоритма разложения  $n$  на множители для достаточно длинных чисел  $n$  (от 1024 бит и больше). При этом, рекомендуется выбирать  $p$  и  $q$  большими случайными простыми числами примерно одинаковой длины. Генерация таких  $n$  — процесс, требующий аккуратности и понимания происходящего. Существует большое количество атак на ключи RSA, которые были сгенерированы ненадлежащим образом. Знание других деталей реализации RSA для этой задачи не понадобится.

Прочитав, что в RSA используют близкие простые числа, Карл реализовал свой алгоритм генерации:

1. Сгенерировать случайное простое число  $p_1$ , состоящее из  $b$  бит.
2. Начиная с  $p_1 + 1$ , перебрать все числа подряд по возрастанию, пока не встретим следующее простое число  $p_2$ .
3. Выдать  $n = p_1 p_2$ .

Поскольку выбирается случайное простое число  $p_1$ , а в среднем расстояние между соседними простыми числами невелико, этот алгоритм достаточно быстро найдет следующее простое число  $p_2$ . Друг Карла Пьер обнаружил, что числа, которые выдает алгоритм Карла, можно быстро разложить на делители. Поэтому Пьер предложил брать не два простых числа, а четыре! Независимо от  $p_1$  мы также выберем случайное  $b$ -битное простое число  $q_1$  и следующее за ним простое число  $q_2$  и возьмем  $n = p_1 p_2 q_1 q_2$ . Однако такой способ генерации тоже оказался уязвим: число  $n$  возможно разложить на множители.

Вам дано число  $n$ , сгенерированное либо изначальным методом Карла с 2 простыми множителями, либо с обновленным методом Пьера с 4 простыми множителями. Разложите его на простые множители.

### Формат входных данных

В первой строке заданы два числа  $b$  и  $k$  ( $4 \leq b \leq 60$ ,  $k = 2$  или  $k = 4$ ). В следующей строке содержится число  $n$  в шестнадцатеричной системе счисления, от старших разрядов к младшим, без ведущих нулей.

Гарантируется, что  $n$  является произведением ровно  $k$  простых множителей, сгенерированных случайно одним из двух методов, описанных в условии. Каждый из этих множителей состоит ровно из  $b$  бит в двоичной системе счисления, все множители различны.

### Формат выходных данных

Выведите  $k$  простых множителей  $n$  в шестнадцатеричной записи без ведущих нулей, по одному в каждой строке.

### Система оценки

Подзадача	Баллы	Ограничения
1	10	$b \leq 8, k = 2$
2	10	$b \leq 8, k = 4$
3	7	$b \leq 15, k = 2$
4	8	$b \leq 15, k = 4$
5	15	$b \leq 30, k = 2$
6	15	$b \leq 30, k = 4$
7	15	$b \leq 60, k = 2$
8	20	$b \leq 60, k = 4$



## Примеры

стандартный ввод	стандартный вывод
4 2 8f	b d
6 4 534ee3	25 29 3b 3d

## Пояснения к примерам

В первом примере  $n = 8f_{16} = 143 = 11 \cdot 13$ .  $b_{16}$  равно числу 11, а  $d_{16}$  равно числу 13. Во втором примере задано число  $n = 5459683$ , что раскладывается в произведение  $37 \cdot 41 \cdot 59 \cdot 61$ .





## Задача E. Black Friday

Ограничение по времени: 1 секунда

Ограничение по памяти: 512 мегабайт

Сегодня — черная пятница. В честь этого Миша решил закупить оборудование для своей майнинг фермы. В магазине действует удивительная акция, если купить видеокарту номер  $i$  в количестве от  $l_i$  до  $r_i$ , включительно ( $1.4 \cdot l_i \leq r_i$ ), то их стоимость будет сильно ниже рыночной. Миша может унести с собой не больше  $s$  видеокарт. Из-за такой щедрости магазина он решил купить как можно больше видеокарт по этой акции. Решить такую задачу Мише не представляется возможным, поэтому он просит вас о помощи. Для каждой видеокарты определите, сколько штук надо купить, либо не покупать вообще, чтобы все видеокарты были куплены по акции, при этом их количество было **максимально возможным** числом не большим  $s$ .

### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $s$  ( $1 \leq n \leq 10^5$ ,  $1 \leq s \leq 10^{13}$ ) — количество различных видеокарт и максимальное число видеокарт, которое Миша может унести с собой.

Следующие  $n$  строк содержат по два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq 10^{13}$ ,  $1.4 \cdot l_i \leq r_i \leq 10^{13}$ ) — минимальное и максимальное количество видеокарт типа  $i$ , которое можно купить по скидке.

### Формат выходных данных

В первой строке выведите единственное целое число  $w$  — максимальное количество видеокарт, которое можно купить по скидке ( $0 \leq w \leq s$ ). Во второй строке выведите через пробел  $n$  целых чисел  $x_i$ .  $i$ -е число равно количеству видеокарт типа  $i$ , которое необходимо купить. Заметим, что либо  $x_i = 0$ , либо  $l_i \leq x_i \leq r_i$ , а также что сумма  $x_i$  равна  $w$ . Если существует несколько решений, максимизирующих  $w$ , выведите любое.

### Система оценки

Подзадача	Баллы	Дополнительные ограничения
1	10	$n \leq 20$
2	14	$n \leq 500$ , $s \leq 2 \cdot 10^5$
3	16	$n \leq 500$
4	29	$n \leq 5000$
5	31	нет дополнительных ограничений

### Пример

стандартный ввод	стандартный вывод
3 20 1 2 10 17 11 16	19 2 17 0

### Пояснение к примеру

В примере из условия выгодно купить видеокарты первого и второго типа. Заметим, что нельзя одновременно купить видеокарты второго и третьего типа, так как тогда нужно будет купить хотя бы  $10 + 11 = 21$  штуку, что больше 20.