

Муниципальный этап всероссийской олимпиады школьников по информатике
Москва, 10 декабря 2017 г.

Задания для 7–8 классов

Продолжительность тура составляет 2 часа (120 минут).

Каждая задача оценивается в 10 баллов. Итоговый балл выставляется как сумма баллов за 4 задачи с лучшим результатом (т.е. для получения максимального балла нужно решить любые 4 задачи). Вы можете отправить на проверку не более 100 решений суммарно по всем задачам 1–7.

Сохраните свой логин и пароль. Вечером вы сможете ознакомиться с результатами проверки своих решений в тестирующей системе, используя свой логин и пароль.

Информация о порядке подачи апелляций будет опубликована на сайте olympiads.ru/moscow

Во время тура в задачах 1–4 производится проверка ответа на соответствие формату, описанному в условии задачи. Если формат записи ответа соответствует условию задачи, решение получает статус «Принято на проверку», иначе решение получает статус «Неправильный формат ответа». После окончания олимпиады будет проверено и оценено последнее принятое на проверку решение по каждой задаче 1–4.

Задача 1. Счастливым билет

Петя едет в автобусе с кондуктором, который отрывает билеты от одной ленты. Билеты на ленте пронумерованы последовательными шестизначными числами. Петя увидел номер билета, который кондуктор оторвал очередному пассажиру, и хочет получить от кондуктора счастливый билет, т.е. такой билет, у которого сумма первых трёх цифр равна сумме последних трёх цифр. Например, билет 123051 является счастливым, потому что $1 + 2 + 3 = 0 + 5 + 1$. Помогите Пете определить, когда ему нужно подойти к кондуктору за билетом, т.е. определите номер ближайшего счастливого билета, который оторвёт кондуктор.

Вам необходимо решить задачу для пяти различных номеров билетов:

125112

265680

398083

412715

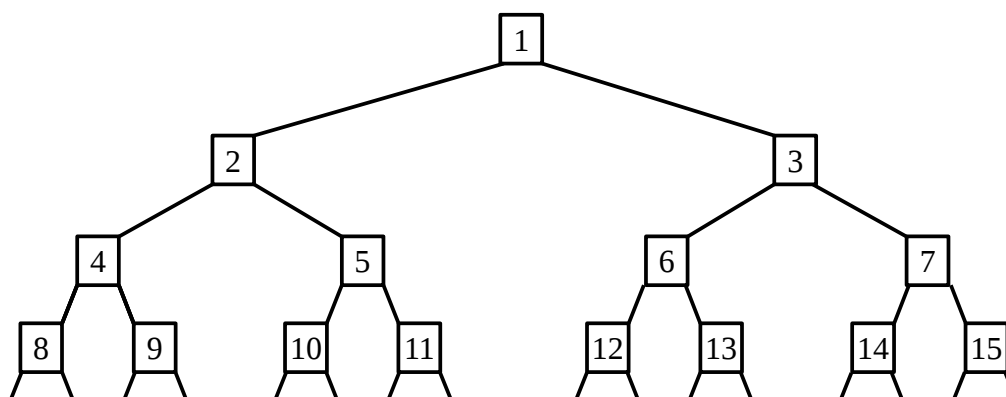
579994

Ответ на эту задачу нужно записать в виде пяти шестизначных чисел, каждое число записано в одной строке, в том же порядке, как они записаны в условии (в первой строке – ответ для первого числа, во второй строке – для второго и т. д.). Если вы не можете дать ответ для какого-то числа, напишите вместо него любое шестизначное число.

Задача 2. Бегущий город

Город состоит из очень большого числа площадей, пронумерованных натуральными числами, расположенными в вершинах двоичного дерева (см. рисунок на следующей странице). Дороги в городе проходят между вершинами дерева от родителя (вершины с номером n) к потомкам (вершинам с номерами $2n$ и $2n + 1$). План начальной части города (только небольшой его части, сам город очень большой) приведён на следующей странице.

Игра «Бегущий город» заключается в том, что участники стартуют на площади номер 1 и должны добраться до контрольных пунктов, размещённых на площадях города. Ваша задача построить пути от площади 1 до пяти контрольных пунктов, размещённых на площадях с номерами 50, 118, 132, 511, 2017.



Ответ на эту задачу следует записать в пяти строках. В первой строке нужно записать маршрут от площади номер 1 до площади номер 50, во второй строке – от площади 1 до площади 118, в третьей строке – от площади 1 до площади 132, в четвёртой строке – от площади 1 до площади 511, в пятой строке – от площади 1 до площади 2017.

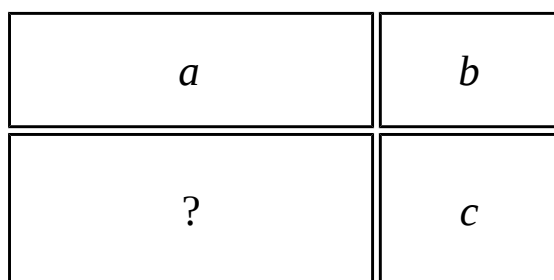
Каждый маршрут записывается в виде последовательности номеров площадей через пробел. Последовательность должна начинаться числом 1 и заканчиваться нужным номером площади. Каждые две соседние площади в одном маршруте должны быть соединены дорогой, в одном маршруте номера площадей не должны повторяться. Например, маршрут от площади 1 до площади 13 нужно записать в виде

1 3 6 13

При сдаче задания проверяется то, что ответ состоит из пяти строк, каждая строка начинается с числа 1, и строки заканчиваются числами 50, 118, 132, 511, 2017 соответственно. Решение получает статус «Принято на проверку», если выполнены эти условия, при этом корректность самого маршрута не проверяется. Если вы не можете найти ответ для какой-то из площадей, напишите в этой строке два числа: 1 и конечный номер площади.

Задача 3. Периметр

В здании был большой конференц-зал в форме прямоугольника. Его разделили на четыре меньших прямоугольных помещения, поставив две перпендикулярные стены (см. рисунок).



Для проведения ремонта необходимо определить периметр каждого из четырёх помещений. Три из четырёх помещений имеют периметр, равный a , b , c (в порядке обхода по часовой стрелке, начиная с левого верхнего угла плана). Определите периметр четвёртого помещения.

Ответом на эту задачу является некоторое выражение, которое может содержать целые числа, переменные a , b и c (записываемые английскими буквами), операции сложения (обозначаются «+»), вычитания (обозначаются «-»), умножения (обозначаются «*»), деления (обозначаются «/») и круглые скобки для изменения порядка действий. Запись вида « $2a$ » для обозначения произведения числа 2 и переменной a неверная, нужно писать « $2 * a$ ».

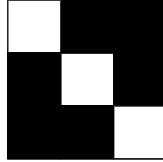
Пример правильного (по форме записи) выражения: $a + (b - c) * 2$.

Задача 4. Черно-белая графика

В этой задаче вам нужно картинку из чёрных и белых пикселей закодировать как можно более короткой строкой по описанным ниже правилам.

Картинка представляет собой прямоугольную таблицу, каждая клеточка которой покрашена в чёрный или белый цвет. Сначала чёрные клеточки обозначаются буквой «B», белые клеточки обозначаются буквой «W». Затем буквы из таблицы записываются подряд в одну строку: сначала первая строка, затем вторая и т. д.

Например, пусть дана следующая картинка:



Обозначим клеточки буквами.

W	B	B
B	W	B
B	B	W

Теперь запишем все буквы в одну строку: «WB²B²WB²W».

Далее эту строку можно сжать, используя следующие правила.

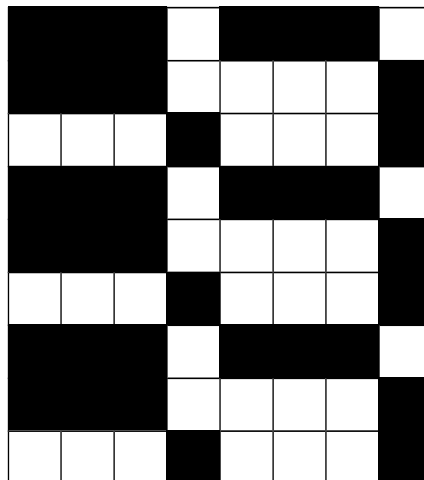
Если перед буквой записано число, то это означает повторение данной буквы указанное число раз. Например, вместо «B²B²» можно написать «3B».

После числа можно написать не одну букву, а последовательность букв в скобках. Например, запись «4(BW)» будет означать последовательность «BWBWBWBW».

Также внутри скобок могут быть записаны не только буквы «B» и «W», но и любые правильно закодированные последовательности, в т.ч. содержащие числа и скобки.

Приведённую выше картинку можно закодировать, например, таким способом: «2(W3B)W».

Дана следующая картинка. Закодируйте её при помощи как можно более короткой последовательности.



Ответ на эту задачу необходимо записать в виде строки, состоящей из букв «B» и «W», чисел и скобок, последовательность должна быть корректной и после распаковки должна соответствовать приведённой картинке. При сдаче задачи проверяется корректность последовательности и то, что в результате распаковки будет получена данная картинка. Если последовательность некорректна или не соответствует картинке, ваше решение получит статус «Неверный формат ответа».

Чем короче будет ваше решение, тем больше баллов вы получите. При подсчёте длины решения учитываются все символы: буквы, цифры и скобки.

Задачи по программированию

Решением задач 5–7 является программа, написанная на одном из языков программирования. Для получения полного балла не обязательно решать задачи 5–7.

Ограничение по времени работы программы в задачах 5–7 – 1 секунда.

Решения оцениваются, только если они выдают правильный ответ на всех примерах входных и выходных данных, приведённых в условии задачи. Проверка решений производится сразу же после отправки, по каждой задаче оценивается решение, набравшее наибольшее число баллов. На странице «Итог» вы можете видеть окончательный балл по задачам 5–7.

Во всех задачах целые числа во входных и выходных данных записываются только цифрами (т.е. недопустимо использование записи 1000000.0 или 1e6 вместо числа 1000000). Каждое число во входных данных записано в отдельной строке.

Сохраните свой логин и пароль. Вечером вы сможете ознакомиться с результатами проверки своих решений в тестирующей системе, используя свой логин и пароль.

Примеры реализации ввода-вывода, документация различных языков программирования, информация о порядке подачи апелляций будут опубликованы на сайте olympiads.ru/moscow

Задача 5. Комета Бармалея

Как известно, комета Бармалея видна с Земли каждые C лет. Любопытно, что это происходит в годы, кратные C , т.е. C , $2 \times C$, $3 \times C$ и т.д. Не каждому суждено увидеть эту комету хотя бы однажды в жизни. Впрочем, находятся счастливые долгожители, заставшие её прилёт даже несколько раз.

Считается, что впервые эту комету увидел и документировал знаменитый средневековый астроном Бармалео Бармалей. В честь него она и получила своё имя. Говорят, за свою долгую жизнь он успел сделать много великих открытий в самых разных областях науки. Однако недавно историки засомневались, правда ли все открытия, которые ему приписываются, Бармалео Бармалей сделал сам. В частности, они заинтересовались, сколько раз за свою жизнь учёный мог видеть комету, названную в его честь.

Бармалео Бармалей родился 1 января в год A и умер 31 декабря в год B . Сколько раз за его жизнь комета была видна с Земли? Мы считаем, что он мог видеть комету, даже будучи младенцем или глубоким стариком, т.е. если она прилетала в год A или B .

Программа получает на вход три целых числа A , B и C ($1 \leq A \leq B \leq 2 \times 10^9$, $1 \leq C \leq 2 \times 10^9$) и должна вывести одно целое число – количество раз, которое комета была видна между годами A и B включительно.

Пример входных и выходных данных

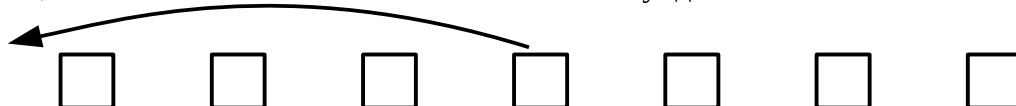
Ввод	Вывод	Примечание
1850 1900 50	2	Комета пролетала около Земли в 1850 и 1900 годах. Бармалео Бармалей застал оба раза.

Система оценивания

Решение, правильно работающее только для случаев, когда все числа не превосходят 10 000, будет оцениваться в 6 баллов.

Задача 6. Переключение окон

Дима – программист, поэтому на его компьютере всегда открыто много окон. Так как у Димы не очень большой монитор, на нём может отображаться только одно окно. В каждый момент времени оконный менеджер хранит список открытых окон, окно с номером 1 отображается на мониторе. Для переключения окон Дима использует сочетание клавиш Alt + Tab. Если удерживать эту кнопку нажатой в течение T секунд, то $T + 1$ -е по счёту окно в текущей нумерации переместится на первую позицию, а относительный порядок остальных окон не изменится. Например, на рисунке ниже показано, что произойдёт с порядком окон, если нажимать на Alt + Tab в течение 3 секунд.



Если держать Alt + Tab $N - 1$ секунду, то первым станет последнее окно из списка. Список открытых окон «заиклен», за последним окном следует первое окно из списка, т. е. если удерживать Alt + Tab нажатым N секунд, то окно, которое было первым в списке, останется на первом месте. Если удерживать Alt + Tab $N + 1$ секунду, на первое место переместится второе по счёту окно и т.д.

В начале рабочего дня любимая среда разработки Димы имела номер M в списке открытых окон. В течение дня Дима K раз использовал сочетание клавиш Alt + Tab. Определите, на какой позиции находится его любимая среда разработки в конце дня.

Первая строка входных данных содержит целое число N , $1 \leq N \leq 10^5$ – количество окон на экране. Вторая строка содержит целое число M , $1 \leq M \leq N$ – номер, который имела любимая среда разработки Димы в начале дня. Третья строка содержит целое число K , $1 \leq K \leq 10^5$ – количество раз, которое Дима нажимал Alt + Tab. В последующих K строках содержатся целые положительные числа, не превосходящие 10^5 – длительность каждого нажатия в секундах.

Программа должна вывести одно целое число – позицию любимой среды Димы в конце рабочего дня.

Пример входных и выходных данных

Ввод	Вывод	Примечание
3 2 3 1 5 2	3	На экране три окна. Пронумеруем окна от 1 до 3 в том порядке, в котором они располагались в начале дня. Димина среда разработки имела номер 2. Дима нажимал на Alt + Tab три раза, продолжительность нажатий была 1, 5 и 2 секунды. Тогда расположение окон после каждого из нажатий будет таким: Нажатие в течение 1 с, второе окно перемещается в начало – 2 1 3. Нажатие в течение 5 с, третье окно перемещается в начало – 3 2 1 Нажатие в течение 2 с, третье окно перемещается в начало – 1 3 2 В результате Димина среда разработки оказалась на месте 3 в списке.

Система оценивания

Решение, правильно работающее только для случаев, когда $1 \leq N \leq 3$, $1 \leq K \leq 3$ и все продолжительности нажатий не превосходят $N - 1$, будет оцениваться в 3 балла.

Решение, правильно работающее только для случаев, когда $1 \leq N \leq 100$ и $1 \leq K \leq 100$, будет оцениваться в 6 баллов.

Задача 7. SNTP

Для того чтобы компьютеры поддерживали актуальное время, они могут обращаться к серверам точного времени SNTP (Simple Network Time Protocol). К сожалению, компьютер не может просто получить время у сервера, потому что информация по сети передаётся не мгновенно: пока сообщение с текущим временем дойдёт до компьютера, оно потеряет свою актуальность. Протокол взаимодействия клиента (компьютера, запрашивающего точное время) и сервера (компьютера, выдающего точное время) выглядит следующим образом:

- 1) Клиент отправляет запрос на сервер и запоминает время отправления A (по клиентскому времени).
- 2) Сервер получает запрос в момент времени B (по точному серверному времени) и отправляет клиенту сообщение, содержащее время B .
- 3) Клиент получает ответ на свой запрос в момент времени C (по клиентскому времени) и запоминает его. Теперь клиент, из предположения, что сетевые задержки при передаче сообщений от клиента серверу и от сервера клиенту одинаковы, может определить и установить себе точное время, используя известные значения A, B, C .

Вам предстоит реализовать алгоритм, с точностью до секунды определяющий точное время для установки на клиенте по известным A, B и C . При необходимости округлите результат до целого числа секунд по правилам арифметики (в меньшую сторону, если дробная часть числа меньше $\frac{1}{2}$, иначе в большую сторону).

Возможно, что, пока клиент ожидал ответа, по клиентскому времени успели наступить новые сутки, однако известно, что между отправкой клиентом запроса и получением ответа от сервера прошло менее 24 часов.

Программа получает на вход три временные метки A, B, C . Каждая временная метка состоит из трёх целых чисел: количества часов, количества минут, количества секунд. То есть первые три строки входных данных содержат числа A_h, A_m, A_s – часы, минуты, секунды момента A по клиентскому времени. Следующие три строки содержат числа B_h, B_m, B_s – часы, минуты, секунды момента B по времени сервера. Следующие три строки содержат числа C_h, C_m, C_s – часы, минуты, секунды момента C по времени клиента.

Программа должна вывести три целых числа: часы, минуты, секунды вычисленного точного времени, которое должен установить себе клиент.

Пример входных и выходных данных

Ввод	Вывод	Примечание
15 1 0 18 9 45 15 1 40	18 10 5	Клиент отправил запрос в 15:01:00 по своим часам, сервер получил запрос в 18:09:45 по своим часам. Клиент получил ответ в 15:01:40, в этот момент точное время будет 18:10:05.

Система оценивания

Решение, правильно работающее только для случаев, когда все три входных времени и ответ на задачу принадлежат одним суткам, будет оцениваться в 6 баллов.