

# Задачи районных олимпиад по информатике в Нижегородской области, 2009—2019

Собраны с разных ресурсов в интернете

Петр Калинин  
[petr@kalinin.nnov.ru](mailto:petr@kalinin.nnov.ru)  
<http://algotprog.ru>

## О формате олимпиад и тестах

До 2016 года включительно олимпиада проводилась в «классическом» формате: в течение тура школьники писали решения, после тура решения проверялись отдельно в каждом районе.

Поэтому в комплекты задач, рассылаемые по районам, до 2016 года входили тесты, и в большинстве случаев они приведены в этом сборнике.

Начиная с 2017 года олимпиада проводится в системе Яндекс.Контест с онлайн-тестированием в течение тура. Поэтому в комплекты задач, рассылаемые по районам, тесты не входят, и их вообще как правило нет в свободном доступе. Поэтому тестов нет и в этом сборнике. С другой стороны, как правило, соответствующие задачи взяты с разных старых олимпиад, поэтому нередко тесты по этим задачам можно найти в интернете. Правда, как показывает опыт, нередко тесты на муниципальном этапе оказываются очень слабыми, и/или ограничение по времени выставляется так, что проходят даже слишком медленные решения.

2019-20

### Задача 1. Супердвоичная система счисления

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Недавно палеонтологи обнаружили останки динозавра *Linhenykusmonodactylus*, у которого на каждой передней конечности было только по одному пальцу. Распространение десятичной системы счисления связывают с количеством пальцев рук у человека. Значит, динозавры пользовались двоичной системой счисления. Точнее, супердвоичной системой, в которой для записи чисел использовались только «цифры» -1, 0 или 1. Супердвоичной записью числа  $n$  динозавры называли представление  $n$  в виде  $2^k * a[k] + \dots + 2^2 * a[2] + 2 * a[1] + a[0]$ , где каждое из чисел  $a[i]$  равно -1, 0 или 1, и  $a[i] * a[i+1] = 0$  для всех  $0 \leq i \leq k - 1$ . Например, число 3 в этой системе записывалось в виде 1 0 -1, так как  $3 = 2^2 * 1 + 2 * 0 + (-1)$ .

Ваша задача – научиться записывать числа в супердвоичной системе динозавров.

#### Формат входного файла

В единственной строке записано целое число  $n$  ( $1 \leq n \leq 10^{18}$ ).

#### Формат выходного файла

Единственная строка содержит последовательность из разделенных пробелом целых чисел  $a[k], \dots, a[1], a[0]$ , образующих запись числа  $n$  в супердвоичной системе счисления. Число  $a[k]$  является первой (слева) цифрой в записи числа  $n$ , а  $a[0]$  – его последней цифрой.

Если таких представлений несколько, выведите любое из них.

#### Пример входных и выходных файлов

1	1
3	1 0 -1

### Задача 2. Популярный рейтинг

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

На конференцию по проблемам в области информационных технологий приехали  $n$  известных программистов и учёных со всего мира. Авторитет конференции зависит от рейтинга участников; рейтинг каждого учёного – это целое положительное число  $r$ , равное количеству его научных публикаций. Число  $r$  считается популярным, если более половины участников конференции имеют рейтинг  $r$ .

Вам необходимо составить программу, которая из данных  $n$  рейтингов учёных определяет популярный.

#### Формат входного файла

В первой строке записано одно число  $n$  – количество участников конференции ( $2 \leq n \leq 10^6$ ).

Во второй строке записаны  $n$  целых положительных чисел из промежутка  $[1; 10^9]$  – рейтинги участников. Гарантируется, что среди них есть популярный рейтинг.

#### Формат выходного файла

Выведите одно целое число – популярный рейтинг участников конференции.

#### Пример входных и выходных файлов

2 1 1	1
5 5 8 5 8 8	8

### Задача 3. Шестерёночки

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Даны  $n$  шестерёнок, некоторые из них соединены между собой. Две сцепленные шестерёнки могут вращаться только в разных направлениях.

Вам необходимо выяснить, может ли вращаться вся система шестерёнок, и если может, указать наименьшее количество шестерёнок, которые нужно заставить вращаться.

#### Формат входного файла

В первой строке записаны два целых числа:  $n$  – количество шестерёнок и  $m$  – количество сцеплений между ними ( $2 \leq n \leq 10^3, 1 \leq m \leq 10^5$ ).

В каждой из следующих  $m$  строк записаны два различных числа  $i$  и  $j$ , которые определяют номера сцепленных шестерёнок. Все шестерёнки пронумерованы целыми числами от 1 до  $n$ .

#### Формат выходного файла

В первой строке запишите одно число  $k$  – наименьшее количество шестерёнок, которые нужно заставить вращаться.

В следующей строке  $k$  целых чисел – номера этих шестерёнок. Если решений несколько, выведите любое из них.

Если запустить все шестерёнки невозможно, выведите -1.

#### Пример входных и выходных файлов

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

#### Пояснение к примеру

В первом примере имеется  $n = 6$  шестерёнок, между ними  $m = 3$  соединения. Все они будут вращаться, если запустить три шестерёнки с номерами 1, 4 и 6.

Во втором примере все шестерёнки вращаться не смогут, поэтому в ответе -1.

#### Задача 4. Сортировочная

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Даны  $n$  целых чисел  $d[1], d[2], \dots, d[n]$  ( $2 \leq n \leq 20$ ). Вам разрешается менять местами любые два числа в этом наборе, если одно из них делится на другое.

Требуется отсортировать исходный набор по возрастанию, переставляя на каждом шаге только два числа.

##### Формат входного файла

В первой строке записано одно целое  $n$  – количество чисел в исходном наборе ( $2 \leq n \leq 20$ ).

Во второй строке записано  $n$  различных целых положительных чисел из промежутка  $[1; 10^9]$ .

Гарантируется, что количество требуемых перестановок не превышает  $10^6$ .

##### Формат выходного файла

В первой строке запишите целое число  $k$  – количество перестановок, которое вам понадобилось для сортировки исходного набора чисел по возрастанию.

В каждой из следующих  $k$  строк запишите по два целых числа из исходного набора, которые меняются местами в соответствии с правилами сортировки (одно из них обязательно должно делиться на другое).

Если решений несколько, выведите любое из них. Если требуемую сортировку сделать невозможно, запишите -1.

2 12 6	1 6 12
2 7 12	0
3 5 4 2	-1

# 2018-19

Задачи были отдельно для 7-8 классов и отдельно для 9-11. Ниже приведены задачи за 9-11 классы, потому что выступать за 7-8 классы нет смысла. Но для справки, за 7-8 классы были даны те же задачи, что и за 9-11 классы, за исключением самой сложной (4-й) задачи и с добавлением одной очень простой.

# 1. Простая математика

Имя входного файла: `sm.in`  
Имя выходного файла: `sm.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Братья Петя и Вася делают успехи в изучении математики. Отец мальчиков даёт им сложные математические задачи, которые братья довольно быстро решают.

Однажды, отец задал новую задачку с простой формулировкой. Требуется найти сумму первых  $N$  членов последовательности, в которой  $i$ -ый элемент равен сумме квадратного корня из  $i$ , округлённого вверх, и кубического корня из  $i$ , округлённого вниз. Т.е.  $i$ -ый элемент последовательности равен

$\text{ceil}(\sqrt{i}) + \text{floor}(\sqrt[3]{i})$ .

Мальчики легко выписали первые члены последовательности и суммировали их. Однако задумались над вопросом: как решить задачу, если  $N$  будет очень большим числом. В этом случае записать все числа последовательности будет непросто.

После некоторых размышлений мальчики решили задачу для больших  $N$ . Но теперь, прежде чем показывать ответы отцу, они хотят убедиться в их правильности. Они предлагают вам решить ту же задачу, чтобы сравнить ответы.

## Формат входного файла

Единственная строка входного файла содержит число  $N$  ( $1 \leq N \leq 10^{12}$ ) – количество элементов последовательности, сумму которых вы должны получить.

## Формат выходного файла

Выходной файл должен содержать одно число – сумму  $N$  первых членов последовательности.

## Примеры входных и выходных файлов

<code>sm.in</code>	<code>sm.out</code>
1	2
4	11

## Пояснение к примерам

Первые члены последовательности равны: 2, 3, 3, 3,...



## 2. Эволюция

Имя входного файла: evo.in  
Имя выходного файла: evo.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Учёные Байтландии давно знают, что ДНК (Дезоксирибонуклеиновая Кислота) содержит генетическую информацию организма. В настоящее время нетрудно извлечь ДНК любого организма, расшифровать этот код и рассмотреть интересующий сегмент для дальнейших исследований. В научных кругах под расшифровкой понимается представление генетического кода в виде строки, состоящей из символов А, С, Т и G. Подобная строка полностью описывает код ДНК.

Учёным предстоит ещё очень много работы, чтобы понять все секреты природы, спрятанные в ДНК. Тем не менее, уже имеются большие успехи в этом направлении. Например, не так давно мир узнал о возможности определения уровня эволюции организма по отдельным участкам кода ДНК. Для этого берётся ДНК и выбирается участок из N нуклеотидов, по которому считается степень сложности. Чем выше степень сложности выбранного сегмента, тем большее количество стадий эволюции прошёл организм.

Степень сложности сегмента ДНК определяется следующим образом:

- Для сегмента из N нуклеотидов рассматриваются все подсегменты из K последовательных нуклеотидов;
- Для каждого подсегмента подсчитывается количество содержащихся в нём нуклеотидов каждого типа (иными словами, вычисляются четыре числа, которые определяют количественную композицию нуклеотидов в подсегменте);
- Степень сложности определяется как количество различных количественных композиций подсегментов. Две композиции считаются одинаковыми, если количества соответствующих нуклеотидов равны в обеих композициях.

После этого открытия учёные начали собирать информацию о степенях сложности ДНК различных организмов. У них уже имеются все необходимые коды ДНК, однако исследование каждой цепочки ДНК вручную занимает очень много времени. Поэтому они попросили вас написать программу, которая позволила бы быстро определять степень эволюции организма.

### Формат входного файла

Первая строка входного файла содержит два целых числа N и K ( $1 \leq N \leq 100\,000$ ) – количество нуклеотидов в сегменте цепочки ДНК и длина подсегментов для рассмотрения. Числа разделены одним пробелом.

Вторая строка содержит N символов, соответствующих нуклеотидной цепочки. Каждый символ – это А, С, Т или G.

### Формат выходного файла

Единственная строка выходного файла должна содержать одно число – степень сложности организма.

### Примеры входных и выходных файлов

evo.in	evo.out
5 1 ACGTA	4
8 2 AACACGTA	5

### Пояснения к примерам

В первом примере имеется 4 различных подсегмента единичной длины.

Во втором примере имеется 7 подсегментов длины 2: AA, AC, CA, AS, CG, GT, TA. Подсегменты AC и CA имеют эквивалентные количественные композиции, поэтому степень сложности всего сегмента равна 5.

### 3. Сортировка

Имя входного файла: `elective.in`  
Имя выходного файла: `elective.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Всеми любимый учитель Мистер Тринидад решил обновить программу своих занятий. Он выяснил, что учащиеся очень интересуются числовыми последовательностями и специальными преобразованиями.

На первом занятии М-р Тринидад сообщил учащимся, что последовательность чисел  $a_1, a_2, \dots, a_N$  называется сортированной, если для любого индекса  $i < N$  выполняется условие  $a_i \leq a_{i+1}$ . Затем учитель объяснил наиболее известные алгоритмы сортировки последовательностей. Т.к. учащиеся довольно быстро усвоили материал занятия, учитель решил рассказать о менее известном методе сортировки.

Суть метода проста: последовательность чисел  $a_1, a_2, \dots, a_N$  делится на  $M$  сегментов. Первый сегмент содержит числа с индексами от 1 до  $k_1$ , второй – от  $k_1 + 1$  до  $k_2$ , ..., последний – от  $k_{M-1} + 1$  до  $k_M$ . Далее, в процессе сортировки, числа не меняют своего положения в пределах сегментов. Сортировка производится перестановкой сегментов целиком.

С целью облегчить понимание этого метода сортировки учитель задал ученикам домашнее задание: разбить последовательность на минимальное количество сегментов таким образом, чтобы можно было применить вышеописанный метод сортировки.

Вы, как самый продвинутый ученик, решили написать программу, решающую задачу для любой числовой последовательности.

#### Формат входного файла

Первая строка входного файла содержит число  $N$  ( $1 \leq N \leq 500\,000$ ) – длина последовательности.

Вторая строка содержит  $N$  целых чисел  $a_i$  ( $1 \leq a_i \leq N$ ) – элементы последовательности. Числа в строке разделяются пробелами.

#### Формат выходного файла

Единственная строка выходного файла должна содержать одно число – минимальное количество сегментов, на которое можно разбить исходную последовательность.

#### Примеры входных и выходных файлов

Input	Output
6 5 6 4 3 1 2	4
3 1 2 1	2

#### Пояснения к примерам

В первом примере последовательность можно разбить на 4 сегмента (5 6), (4), (3) and (1 2). Сегмент (1 2) должен быть размещён первым, (3) – вторым, (4) – третьим, а (5 6) – последним. Таким образом последовательность будет отсортирована.

Во втором примере последовательность делится на два сегмента: (1 2) и (1).

## 4. Статистическое шифрование

Имя входного файла: encrypt.in  
Имя выходного файла: encrypt.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В Институте Информационной Безопасности (ИИБ) Бейтландии учёные разрабатывают новые способы защиты передаваемой и хранимой информации.

Недавно в ИИБ разработан метод безопасной передачи числовых последовательностей, названный «Статистическое шифрование». Идея метода проста: числовая последовательность подвергается процессу шифрования, в итоге получается две последовательности: шифр и ключ.

Сам процесс шифрования держится в секрете, но члены ИИБ хотят нанять программиста, который бы реализовал декодер. Декодер должен по зашифрованной последовательности и ключу получать исходную (незашифрованную) числовую последовательность.

В ИИБ вам сообщили детальную информацию о декодере. На вход поступают следующие данные:

3. зашифрованная числовая последовательность;
4. ключ шифрования  $k_i$  ( $1 \leq i \leq K$ ), который тоже является числовой последовательностью;
5. параметр  $M$ , который задаёт размер обрабатываемых блоков последовательности. Блок — это непрерывный участок последовательности из  $M$  элементов.

Для восстановления  $i$ -го элемента исходной последовательности нужно вычислить статистику  $k_i$ -го порядка для каждого блока: число, которое будет стоять на  $k_i$ -м месте в блоке после его сортировки. Элемент в позиции  $i$  исходной последовательности равен максимальному значению из вычисленных статистик соответствующего порядка.

Напишите декодер, который будет расшифровывать зашифрованную информацию.

### Формат входного файла

Первая строка входного файла содержит три числа  $N, M, K$  ( $1 \leq N \leq 250\,000$ ,  $1 \leq K \leq M \leq N$ ) — соответственно, количество элементов в зашифрованной последовательности, параметр декодера и количество элементов в исходной последовательности.

Вторая строка содержит  $N$  различных целых чисел  $a_i$  ( $1 \leq a_i \leq 250\,000$ ) — зашифрованная последовательность.

Последняя строка содержит  $K$  различных целых чисел  $k_i$  ( $1 \leq k_i \leq M$ ) — элементы ключа шифрования.

Все числа в строках разделены одинарными пробелами.

### Формат выходного файла

Выходной файл должен содержать  $K$  целых чисел, представляющих исходную последовательность.

### Примеры входных и выходных файлов

Input	Output
5 2 2 1 5 3 2 6 1 2	3 6
4 3 2 5 7 3 4 3 2	7 5

Пояснения к примерам

К первому примеру: Рассмотрим все блоки последовательности: (1, 5), (5, 3), (3, 2) и (2, 6). Первый элемент исходной последовательности равен максимуму из минимумов в блоках (т.е. 3). Второй элемент — наибольшему из максимумов в блоках (т.е. 6).

Ко второму примеру: Здесь имеется только два блока: (5, 7, 3), (7, 3, 4). Первый элемент равен наибольшему из максимумов в блоках (т.е. 7). Второй — наибольшему из медиан (вторые элементы по значимости) в блоках (т.е. 5).

# 2017-18

Задачи были отдельно для 7-8 классов и отдельно для 9-11. Ниже приведены задачи за 9-11 классы, потому что выступать за 7-8 классы нет смысла. Но для справки, за 7-8 классы были даны те же задачи, что и за 9-11 классы, за исключением самой сложной (4-й) задачи и с добавлением одной очень простой.

Надо еще отметить, что результаты олимпиады оказались очень плотными в районе 200+ers баллов: 238 баллов — это было 8-10 место в общем зачете (по всем классам), а 200 баллов — это 83-95 место. Поэтому проходные баллы оказались примерно 210-214 по всем классам.

**Задача 1. Урок физкультуры**

Имя входного файла: gym.in

Имя выходного файла: gym.out

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

На уроке физкультуры тренер Андрей Сергеевич выстраивает учеников в одну шеренгу. В шеренге сначала идут мальчики, а потом девочки. При этом мальчики в шеренге стоят по убыванию роста, аналогично девочки тоже стоят по убыванию роста. Таким образом, следом за самым низким мальчиком стоит самая высокая девочка.

Андрея Сергеевича заинтересовал вопрос, какое максимальное различие в росте двух стоящих рядом учеников. Напишите программу, которая поможет Андрею Сергеевичу ответить на этот важный для него вопрос.

**Формат входного файла**Первая строка содержит целое число  $n$  – число учеников в классе ( $2 \leq n \leq 50$ ).

Следующие  $n$  строк содержат по два целых числа каждая:  $a_i$  и  $h_i$  – пол и рост в сантиметрах  $i$ -го ученика ( $a_i$  равно 0 или 1,  $100 \leq h_i \leq 200$ ). Значение  $a_i=0$  означает, что  $i$ -й ученик – мальчик, а значение  $a_i=1$  означает, что  $i$ -й ученик – девочка.

**Формат выходного файла**

Выведите одно число – максимальное различие в росте стоящих рядом учеников после того, как они выстроятся в шеренгу на уроке физкультуры.

**Пример входных и выходных файлов**

6	22
0 120	
1 130	
1 142	
1 115	
0 145	
0 134	

**Задача 2. Пятница, 13-е**

Имя входного файла: friday.in

Имя выходного файла: friday.out

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Календарь жителей планеты Плюк состоит из  $N$  месяцев, каждый месяц состоит ровно из 30 дней, неделя состоит из 7 дней. Особо несчастливыми на планете Плюк считается 13-е число месяца, если оно выпадает на пятницу.

Известно, что Новый год на планете Плюк начался в  $k$ -й по счету день недели (1-й день недели – понедельник, 2-й – вторник, 3-й – среда, ..., 7-й – воскресенье).

Определите, сколько в этом году на планете Плюк будет особо несчастливых пятниц, 13-е.

**Формат входного файла**

Программа получает на вход два натуральных числа, записанных в отдельных строках. Первое число – количество месяцев в календаре планеты Плюк  $N$ , не превосходящее  $10^9$ . Второе число – номер дня недели, на который приходится первое число первого месяца нового года, может принимать значения от 1 до 7.

**Формат выходного файла**

Программа должна вывести единственное целое число – количество несчастливых дней в этом году.

**Пример входных и выходных файлов**

12	2
1	

**Пояснение к примеру**

На 13-е число будут приходиться пятницы четвертого и одиннадцатого месяцев.

### Задача 3. Сумма минимумов

Имя входного файла: summin.in

Имя выходного файла: summin.out

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

У Саши есть блокнот, состоящий из  $n$  листочков, пронумерованных от 1 до  $n$ . На  $i$ -м листочке написано целое число  $a_i$ .

Аня собирается разорвать блокнот на  $k$  частей, для этого она выбирает  $k-1$  число  $1 \leq r_1 < r_2 < \dots < r_{k-1} < n$  и разрывает блокнот так, что листки с 1 по  $r_1$ -й оказываются в первой части, листки с  $(r_1+1)$ -го по  $r_2$ -й оказываются во второй части, и т.д., последняя  $k$ -я часть содержит листки с  $(r_{k-1} + 1)$ -го по  $n$ -й.

После того, как Аня разорвет блокнот, Саша найдет минимальное число в каждой из получившихся частей и сложит их. Аня хочет разорвать блокнот таким образом, чтобы получившаяся сумма была как можно больше. Помогите ей выбрать способ разорвать блокнот, чтобы максимизировать сумму минимальных значений.

#### Формат входного файла

Первая строка ввода содержит два числа:  $n$  и  $k$  ( $2 \leq k \leq n \leq 300$ ).

Вторая строка содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

#### Формат выходного файла

На первой строке выведите максимальное значение суммы, которое удастся достичь Ане.

На второй строке выведите значения  $r_1, r_2, \dots, r_{k-1}$ , которые ей необходимо выбрать. Если вариантов разорвать блокнот, чтобы максимизировать искомую сумму несколько, выведите любой из них.

#### Пример входных и выходных файлов

10 5	27
1 10 2 8 9 3 5 4 7 6	3 4 5 8

#### Пояснение к примеру

В приведенном примере Аня разорвала блокнот на части  $[1, 10, 2]$ ,  $[8]$ ,  $[9]$ ,  $[3, 5, 4]$  и  $[7, 6]$ .

Искомая сумма равна  $1 + 8 + 9 + 3 + 6 = 27$ .

### Задача 4. Урюк

Имя входного файла: apricot.in

Имя выходного файла: apricot.out

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

В давние времена Золотая Орда ежегодно собирала дань золотыми монетами. Известный крымский хан Гирей решил схитрить: выплачивая дань из  $N$  золотых монет, он подложил среди них одну фальшивую – более легкую монету. Об этом донесли казначею Золотой Орды. Для обнаружения подделки он решил использовать магические весы, работающие на урюке.

На чаши магических весов кладутся две кучи монет. Весы устанавливают, совпадает или различается вес этих куч. При этом, если кучи имеют разный вес, то весы указывают, какая из куч легче.

При совпадении веса обеих куч весы требуют  $R$  плодов урюка, а при несовпадении –  $U$  плодов. Казначей, сам любитель урюка, хочет и фальшивую монету обнаружить, и сэкономить на урюке.

Требуется написать программу, которая по заданному количеству монет  $N$ , при условии, что только одна из них легче других, укажет минимальное количество урюка, с помощью которого эта фальшивая монета гарантированно будет обнаружена.

#### Формат входного файла

Во входном файле в единственной строке находятся три целых числа  $N, R$  и  $U$  ( $2 \leq N \leq 1\,000\,000$ ,  $1 \leq R, U \leq 1\,000\,000$ ), где  $N$  – количество монет,  $R$  – количество плодов урюка, затрачиваемых в случае совпадения веса куч монет,  $U$  – количество плодов урюка, затрачиваемых в случае их различия. Все числа разделены пробелом.

#### Формат выходного файла

Выходной файл должен содержать одно число – минимальное количество урюка, с помощью которого гарантированно будет обнаружена фальшивая монета.

#### Пример входных и выходных файлов

4 3 1	2
3 3 1	3
15 2 3	8
10 2 1	3

# 2016-17

В 2016 году и ранее олимпиада проводилась не в Яндекс.Контесте, а с тестированием решений после тура.

В 2016-17 году задания были отдельно по 9 классам и отдельно по 10-11 классам.

# 2016-17 год, 8-9 классы

## Речные гонки

Имя входного файла: `river.in`  
Имя выходного файла: `river.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Егор и Петр участвуют в речной гонке на лодках: участники одновременно стартуют в пункте **A** и должны проплыть против течения реки в пункт **B**. Тот, кто приплывет первым, объявляется победителем. В результате жеребьевки Егору и Петру выпало участвовать в одном заплыве. Нам известны скорости движения лодок ребят и скорость течения реки. Ваша задача – определить, кто же из них победит.

### Формат входного файла

В первой строке ввода содержатся три целых числа: **V1** – скорость лодки Егора, **V2** – скорость лодки Петра и **V3** – скорость течения реки ( $0 \leq V1, V2, V3 \leq 10^6$ ).

### Формат выходного файла

Выведите одну строку, содержащую следующий текст:

1. "EGOR", если первым приплывет Егор;
2. "PETR", если первым приплывет Петр;
3. "RIVER", если победителя определить не удалось.

### Пример входных и выходных файлов

<code>river.in</code>	<code>river.out</code>
10 5 1	EGOR
4 5 2	PETR
6 6 2	RIVER

## Сеть

Имя входного файла: `network.in`  
Имя выходного файла: `network.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Для проведения олимпиады организаторы планируют объединить компьютеры участников в сеть. Из сетевого оборудования в наличии есть  $N$  коммутаторов и неограниченное количество сетевых кабелей. Коммутатор с номером  $i$  ( $1 \leq i \leq n$ ) характеризуется числом  $a_i$  — количеством портов в этом коммутаторе.

Организаторы могут соединить кабелем либо два коммутатора, либо два компьютера, либо коммутатор и компьютер. Каждый коммутатор может быть соединен кабелями не более чем с  $a_i$  устройствами (коммутаторами или компьютерами), каждый компьютер — не более чем с одним.

Два компьютера могут обмениваться данными, если от одного из них до другого можно добраться по кабелям, возможно, пройдя при этом по цепочке из коммутаторов. Организаторы хотят построить сеть таким образом, чтобы каждые два компьютера могли обмениваться данными.

Какое максимальное количество компьютеров организаторы могут объединить в сеть, используя имеющиеся коммутаторы?

### Формат входного файла

В первой строке входного файла находится одно число  $N$  — количество коммутаторов, имеющихся у организаторов ( $0 \leq N \leq 10^5$ ).

Во второй строке файла находится  $N$  чисел  $a_i$  — количество портов в коммутаторе с номером  $i$  ( $1 \leq a_i \leq 10^9$ ,  $1 \leq i \leq N$ ).



### Формат выходного файла

Выведите единственное число — максимальное количество компьютеров, которое удастся объединить в сеть, используя имеющиеся коммутаторы.

#### Пример входных и выходных файлов

network.in	network.out
3 10 4 5	15
2 1 10	10
2 3 10	11

### Большое число

Имя входного файла: bignum.in  
Имя выходного файла: bignum.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дано целое число  $N$ , состоящее из четного количества десятичных цифр. Над ним последовательно производятся следующие действия:

- цифры числа разделяются на две равные половины;
- левая и правая половины разворачиваются, то есть порядок следования цифр меняется на противоположный;
- аналогичные действия выполняются для частей числа без первой и последней цифры, и так далее.

Когда останется последняя цифра первой половины числа и первая — второй, процесс останавливается, так как разворачивать их не имеет смысла.

Рассмотрим пример. Пусть  $N = 1234567890$ . Тогда в процессе выполнения указанных действий будет получена следующая цепочка: 5432109876, 5123478906, 5143298706, 5142389706. Ваша задача — узнать результат последовательности указанных преобразований.

### Формат входного файла

Входной файл содержит единственное число  $N$  (не более 10000 цифр). Допускаются нули в начале записи числа.

### Формат выходного файла

Выходной файл должен содержать единственное число — результат применения всех действий.

#### Пример входных и выходных файлов

bignum.in	bignum.out
1234567890	5142389706
000123	000231
012039	201390

### Ох, уж эти скобки

Имя входного файла: brackets.in  
Имя выходного файла: brackets.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Математическое выражение записано в виде произведения:

$$(\pm a_2 x^2 \pm a_1 x \pm a_0) \cdot (\pm b_2 x^2 \pm b_1 x \pm b_0) \cdot (\pm c_2 x^2 \pm c_1 x \pm c_0) \dots$$

Внутри каждой из N скобок произведения находится выражение вида:

$$\pm a_2 x^2 \pm a_1 x \pm a_0,$$

где хотя бы один из коэффициентов  $a_i$  не равен нулю ( $b_i$ ,  $c_i$  и т. д., аналогично). Требуется составить программу, которая перемножает выражения в скобках и выводит полученную функцию в виде многочлена с приведенными по степеням  $x$  слагаемыми, то есть в виде:

$$\pm q_{2N} x^{2N} \pm q_{2N-1} x^{2N-1} \pm \dots \pm q_3 x^3 \pm q_2 x^2 \pm q_1 x \pm q_0.$$

### Формат входного файла

В первой строке входного файла находится число  $N$  ( $1 \leq N \leq 6$ ).

Во второй строке находится выражение из  $N$  пар скобок. Внутри каждой пары скобок находится выражение в виде « $\pm a_2 x^2 \pm a_1 x \pm a_0$ », где « $\pm$ » — это или знак «+», или знак «-». Значение каждого из коэффициентов  $a_i$ ,  $b_i$ ,  $c_i$  и т.д. не превышает 10. Выражение составлено по следующим правилам:

- Всё выражение записывается, начиная со старшей степени переменной по убыванию степеней.
- Если какой-то коэффициент равен нулю, то этот коэффициент и соответствующий ему  $x$  опускаются в записи вместе с арифметическим знаком. Исключением является случай, когда все коэффициенты равны нулю. В этом случае вместо всего выражения указывается единственный коэффициент 0.
- Если  $a_i = \pm 1$  и  $i > 0$ , то единица перед соответствующим ему  $x$  не ставится.
- Если первый отличный от нуля коэффициент положителен, то знак «+» перед ним опускается.
- В выражении отсутствуют пробельные символы (пробел, табуляция) и знаки умножения.

### Формат выходного файла

В первой строке выходного файла выведите результат раскрытия скобок в исходном выражении в следующем формате:

$$\pm q_{2N} x^{2N} \pm q_{2N-1} x^{2N-1} \pm \dots \pm q_1 x \pm q_0.$$

Формат выражения должен полностью соответствовать описанию для входного файла. Скобки вокруг степеней ставить не нужно, они приведены здесь только для читабельности.

### Пример входных и выходных файлов

brackets.in	brackets.out
1 (3x^2+2x-1)	3x^2+2x-1
2 (4x^2+3x+5)(2x^2+4x+1)	8x^4+22x^3+26x^2+23x+5
3 (-x+7)(x)(x^2+x+1)	x^4+6x^3+6x^2+7x
6 (1)(2)(3)(4)(5)(6)	720

# 2016-17 год, 10-11 класс

## Большое число

Имя входного файла: `bignum.in`  
Имя выходного файла: `bignum.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дано целое число  $N$ , состоящее из четного количества десятичных цифр. Над ним последовательно производятся следующие действия:

1. цифры числа разделяются на две равные половины;
2. левая и правая половины разворачиваются, то есть порядок следования цифр меняется на противоположный;
3. аналогичные действия выполняются для частей числа без первой и последней цифры, и так далее.

Когда останется последняя цифра первой половины числа и первая — второй, процесс останавливается, так как разворачивать их не имеет смысла.

Рассмотрим пример. Пусть  $N = 1234567890$ . Тогда в процессе выполнения указанных действий будет получена следующая цепочка: 5432109876, 5123478906, 5143298706, 5142389706. Ваша задача — узнать результат последовательности указанных преобразований.

### Формат входного файла

Входной файл содержит единственное число  $N$  (не более 10000 цифр). Допускаются нули в начале записи числа.

### Формат выходного файла

Выходной файл должен содержать единственное число — результат применения всех действий.

### Пример входных и выходных файлов

<code>bignum.in</code>	<code>bignum.out</code>
1234567890	5142389706
000123	000231
012039	201390

## Ох, уж эти скобки

Имя входного файла: `brackets.in`  
Имя выходного файла: `brackets.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Математическое выражение записано в виде произведения:

$$(\pm a_2 x^2 \pm a_1 x \pm a_0) \cdot (\pm b_2 x^2 \pm b_1 x \pm b_0) \cdot (\pm c_2 x^2 \pm c_1 x \pm c_0) \dots$$

Внутри каждой из  $N$  скобок произведения находится выражение вида:

$$\pm a_2 x^2 \pm a_1 x \pm a_0,$$

где хотя бы один из коэффициентов  $a_i$  не равен нулю ( $b_i$ ,  $c_i$  и т. д., аналогично). Требуется составить программу, которая перемножает выражения в скобках и выводит полученную функцию в виде многочлена с приведенными по степеням  $x$  слагаемыми, то есть в виде:

$$\pm q_{2N} x^{2N} \pm q_{2N-1} x^{2N-1} \pm \dots \pm q_3 x^3 \pm q_2 x^2 \pm q_1 x \pm q_0.$$

### Формат входного файла

В первой строке входного файла находится число  $N$  ( $1 \leq N \leq 6$ ).

Во второй строке находится выражение из  $N$  пар скобок. Внутри каждой пары скобок находится выражение в виде  $\pm a_2 x^2 \pm a_1 x \pm a_0$ , где « $\pm$ » — это или знак «+», или знак «-». Значение каждого из коэффициентов  $a_i$ ,  $b_i$ ,  $c_i$  и т. д. не превышает 10. Выражение составлено по следующим правилам:

5. Всё выражение записывается, начиная со старшей степени переменной по убыванию степеней.
6. Если какой-то коэффициент равен нулю, то этот коэффициент и соответствующий ему  $x$  опускаются в записи вместе с арифметическим знаком. Исключением является случай, когда все коэффициенты равны нулю. В этом случае вместо всего выражения указывается единственный коэффициент 0.

7. Если  $a_i = \pm 1$  и  $i > 0$ , то единица перед соответствующим ему  $x$  не ставится.
8. Если первый отличный от нуля коэффициент положителен, то знак «+» перед ним опускается.
9. В выражении отсутствуют пробельные символы (пробел, табуляция) и знаки умножения.

#### Формат выходного файла

В первой строке выходного файла выведите результат раскрытия скобок в исходном выражении в следующем формате:

$$\pm q_{2N}x^{(2N)} \pm q_{2N-1}x^{(2N-1)} \pm \dots \pm q_1x \pm q_0.$$

Формат выражения должен полностью соответствовать описанию для входного файла. Скобки вокруг степеней ставить не нужно, они приведены здесь только для читабельности.

#### Пример входных и выходных файлов

brackets.in	brackets.out
1 (3x <sup>2</sup> +2x-1)	3x <sup>2</sup> +2x-1
2 (4x <sup>2</sup> +3x+5)(2x <sup>2</sup> +4x+1)	8x <sup>4</sup> +22x <sup>3</sup> +26x <sup>2</sup> +23x+5
3 (-x+7)(x)(x <sup>2</sup> +x+1)	x <sup>4</sup> +6x <sup>3</sup> +6x <sup>2</sup> +7x
6 (1)(2)(3)(4)(5)(6)	720

## Разбиение числа

Имя входного файла: `decomp.in`  
Имя выходного файла: `decomp.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Факториалом числа  $n$  называется произведение  $n! = 1 \cdot 2 \cdot \dots \cdot n$  при  $n > 0$  и  $n! = 1$  при  $n = 0$ .

Количество сочетаний из  $n$  элементов по  $k$  определяется следующим образом:

$$\binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!}, & 0 \leq k \leq n; \\ 0, & k > n. \end{cases}$$

В математике такие числа называются также биномиальными коэффициентами. Требуется представить заданное число  $P$  в виде суммы трех биномиальных коэффициентов:

$$P = \binom{a}{1} + \binom{b}{2} + \binom{c}{3}, 0 \leq a < b < c.$$

#### Формат входного файла

Входной файл содержит единственное число  $P$  ( $1 \leq P \leq 10^{18}$ ).

#### Формат выходного файла

В выходной файл выведите искомые числа  $a, b, c$  ( $0 \leq a < b < c$ ), разделённые пробелами. Если задача не имеет решения, выведите три нуля.

#### Пример входных и выходных файлов

decomp.in	decomp.out
42	1 4 7
31	1 5 6

## НОК

Имя входного файла: `lcm.in`  
Имя выходного файла: `lcm.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Наименьшим общим кратным (НОК) нескольких чисел называется наименьшее натуральное число, которое делится на каждое из этих чисел. Заданы два числа  $N$  и  $K$ . Требуется найти набор из  $N$  различных натуральных чисел, наименьшее общее кратное которых равняется  $K$ . Среди всех этих чисел не должно быть единицы и самого числа  $K$ .

#### Формат входного файла

В первой строке входного файла записаны через пробел два числа  $N$  и  $K$  ( $1 \leq N \leq 1000, 1 \leq K \leq 10^9$ ).

Формат выходного файла

В выходной файл в первой строке выведите искомый набор из N чисел, разделённых пробелами. Если Вы смогли найти несколько наборов, то выведите любой из них.

Если требуемого набора не существует, тогда выведите -1.

Пример входных и выходных файлов

lcm.in	lcm.out
2 14	2 7
12 20736	3 9 27 81 256 128 64 32 16 8 4 2
17 42	-1
7 123456	2 3 4 6 30864 41152 61728

Тесты 2016-17

Речные гонки (100 баллов)

№	Исходные данные	Правильный ответ
1	10 5 1	EGOR
2	4 5 2	PETR
3	6 6 2	RIVER
4	100 1 2	EGOR
5	4 50 10	PETR
6	3 3 3	RIVER
7	10 12 20	RIVER
8	1000000 1000000 100	RIVER
9	100 1000000 10	PETR
10	1000000 100000 10000	EGOR

По 10 баллов за каждый успешно пройденный тест.

Сеть (100 баллов)

№	Исходные данные	Правильный ответ
1	0	2
2	1 10	10
3	2 10 7	15
4	15 1 1 1 1 1 1 1 1 1 1 1 1 1	2
5	4 10 5 6 6	21
6	5 8 6 10 10 10	36
7	15 1 2 1 2 1 2 1 2 1 2 1 2 1	2
8	6 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000	5999999990
9	7 3336772 536366 1 4446832 2 35789893 3	44109858
10	8 2 536366 1 1 2 2 3 3	536368

По 10 баллов за каждый успешно пройденный тест.

Большое число (сложность 100 баллов)

№	Исходные данные	Правильный ответ
1	31	31
2	1234	2143
3	4887009903	0478890930
4	50445572516225079820935171007348 16696149613177219987	1570145435950722859176025294611639167671281493978 070
5	72172199516845062641563483807757 44729281278734623998541087992603 880616992417093840	2712812792217949457156787405803682463461560239898 708601164959829491372069433874807

По 10 баллов за успешно пройденные тесты 1 и 2.  
20 баллов за успешно пройденный тест 3.  
По 30 баллов за успешно пройденные тесты 4 и 5.

**Ох, уж эти скобки (100 баллов)**

№	Исходные данные	Правильный ответ
1	1 (x+1)	x+1
2	2 (x <sup>2</sup> +x+1)(x <sup>2</sup> +x+1)	x <sup>4</sup> +2x <sup>3</sup> +3x <sup>2</sup> +2x+1
3	3 (9x <sup>2</sup> +9x+10)(9x <sup>2</sup> +9x+9) (9x <sup>2</sup> +10x+10)	729x <sup>6</sup> +2268x <sup>5</sup> +4698x <sup>4</sup> +5679x <sup>3</sup> +5040x <sup>2</sup> +2610x+900
4	4 (-8x <sup>2</sup> +5x+9)(-x <sup>2</sup> -6x+9)(7x <sup>2</sup> -5x-10)(9x <sup>2</sup> +3x+7)	504x <sup>8</sup> +2517x <sup>7</sup> -8473x <sup>6</sup> -831x <sup>5</sup> +8180x <sup>4</sup> +2765x <sup>3</sup> +3819x <sup>2</sup> -4635x-5670
5	5 (9x <sup>2</sup> +4x-1)(8x+3)(-4x <sup>2</sup> -9x-6)(-2x <sup>2</sup> -3x-1)(-x <sup>2</sup> +9x-5)	-576x <sup>9</sup> +2552x <sup>8</sup> +15910x <sup>7</sup> +26345x <sup>6</sup> +14596x <sup>5</sup> -4255x <sup>4</sup> -7481x <sup>3</sup> -2160x <sup>2</sup> +123x+90
6	6 (10x <sup>2</sup> -2x-5)(2x <sup>2</sup> +x)(7x <sup>2</sup> -10x+7)(-9x <sup>2</sup> +x-8)(9x <sup>2</sup> -5x+1)(4x <sup>2</sup> +10x-10)	-45360x <sup>12</sup> -31968x <sup>11</sup> +235988x <sup>10</sup> -370760x <sup>9</sup> +300162x <sup>8</sup> -99962x <sup>7</sup> -86718x <sup>6</sup> +153360x <sup>5</sup> -61382x <sup>4</sup> -16510x <sup>3</sup> +14430x <sup>2</sup> -2800x
7	4 (-8x <sup>2</sup> +5x+9)(-x <sup>2</sup> -6x+9)(0) (9x <sup>2</sup> +3x+7)	0
8	2 (-2x <sup>2</sup> +x+2)(-2x <sup>2</sup> -x+1)	4x <sup>4</sup> -7x <sup>2</sup> -x+2
9	5 (1)(x <sup>2</sup> )(x <sup>2</sup> )(x <sup>2</sup> +1)(x+1)	x <sup>7</sup> +x <sup>6</sup> +x <sup>5</sup> +x <sup>4</sup>
10	5 (-x-1)(x <sup>2</sup> )(-x <sup>2</sup> -x)(x <sup>2</sup> )(-x <sup>2</sup> -x-1)	-x <sup>9</sup> -3x <sup>8</sup> -4x <sup>7</sup> -3x <sup>6</sup> -x <sup>5</sup>

По 10 баллов за каждый успешно пройденный тест.

**Разбиение числа (100 баллов)**

№	Исходные данные	Правильный ответ
1	2	0 0 0
2	3	1 2 3
3	53803	31 53 69
4	9242384	193 222 382
5	979763401	1263 1593 1805
6	68057018561	3151 7162 7419
7	8314493051849	18792 26819 36813
8	823282321691624	42740 98208 170308
9	77767858026820071	411949 607499 775623
10	1000000000000000000	378767 1399103 1817121

По 10 баллов за каждый успешно пройденный тест.

**НОК (100 баллов)**

№	Исходные данные	Пример правильного ответа
1	4 90	2 45 3 30
2	3 68	34 4 2
3	10 6592	3296 64 2 4 1648 8 824 16 412 32
4	7 1024	-1
5	3 42	2 3 7
6	3 210	-1
7	5 67264	33632 64 2 4 16816
8	11 6343552	3171776 128 2 4 1585888 8 792944 16 396472 32 198236
9	10 71791232	35895616 128 2 4 17947808 8 8973904 16 4486952 32
10	7 130856528	65428264 11896048 2 4 32714132 8 16357066

По 10 баллов за каждый успешно пройденный тест.

**В данной задаче возможны другие правильные решения. Для проверки правильности полученного результата можно воспользоваться функцией MS Excel НОК() или любым сервисом, обеспечивающим получение НОК нескольких чисел.**

**2015-16**

## Текст муниципальной (районной/городской) олимпиады по информатике 2015–2016 учебного года

**Примечание:** в 2015 году очень много школьников, особенно в 10 и 11 классах, набрали на районной олимпиаде полный балл (400). Это обозначает, что задачи и/или тесты были на самом деле довольно простыми. (П. Калинин)

### «Хоккей» – 100 баллов

Финал чемпионата проводился по новой системе: две команды, вышедшие в него должны сыграть между собой **n** матчей ( $n \leq 16$ ). За победу в каждом матче команда получает 2 очка, за ничью – 1 очко, за поражение – не получает. При равенстве очков в **n** матчах победитель определяется по лучшей разности забитых и пропущенных шайб, а если эта разность у обеих команд равна нулю, то обе команды считаются победителями

#### Входные данные

Входной файл **task1.in** содержит в первой строке число матчей **n**, а в каждой из последующих **n** строк находится по два целых числа **a** и **b** – итоговый счет в соответствующем матче. **a** – количество шайб, забитых первой командой, **b** – количество шайб, забитых второй командой. ( $0 \leq a, b \leq 100$ ).

#### Выходные данные

Выходной файл **task1.out** должен содержать номер выигравшей команды, в случае равенства очков и разности шайб следует вывести 0.

#### Примеры:

task1.in	task1.out
4 6 7 3 5 9 11 12 12	2
3 14 5 15 8 2 20	1
2 5 6 6 5	0

### «Эксперимент» – 100 баллов

В лаборатории проводятся эксперименты. В журнале учета фиксируется время начала эксперимента и его окончания. Необходимо по этим показателям определить его продолжительность. Эксперимент длится менее суток

#### Входные данные

В первой строке входного файла **task2.in** записано время начала эксперимента в формате ЧЧ:ММ:СС (с ведущими нулями), а во второй строке время его окончания в том же формате.

#### Выходные данные

В выходной файл **task2.out** следует вывести продолжительность эксперимента в том же формате.

#### Примеры:

task2.in	task2.out
01:01:01 12:05:06	11:04:05
01:01:01 02:01:00	0:59:59
23:59:59 00:00:00	00:00:01



## «Числа» – 100 баллов

Назовем натуральное число подходящим, если

10. в его записи используются только цифры 1, 2, 3 и 4.

11. соседние цифры этого числа отличаются на единицу.

Подсчитать число подходящих  $n$ -значных чисел ( $1 \leq n \leq 40$ ).

### Входные данные

Исходный файл **task3.in** содержит в единственной строке натуральное число  $n$  – число знаков.

### Выходные

Выходной файл **task3.out** должен содержать натуральное число подходящих  $n$ -значных чисел.

Ваша программа должна

- прочитать из файла **task3.in** исходное число  $n$ ;
- подсчитать и вывести в файл **task3.out** количество подходящих  $n$ -значных чисел (возможно длинное целое).

Пример:

task3.in	task3.out
3	10

Примечание. Имеются в виду числа 121, 123, 212, 232, 234, 321, 323, 324, 432, 434.

## «Покраска» – 100 баллов.

На прямой окрасили  $n$  отрезков. Известны координата  $L[i]$  левого конца отрезка и координата  $R[i]$  правого конца  $i$ -го отрезка для  $i=1, \dots, n$ . Найти длину самой большой образовавшейся окрашенной части прямой.

### Входные данные

Исходный файл **task4.in** содержит в первой строке натуральное число  $n$  – число отрезков ( $n \leq 10000$ ). Следующие  $n$  строк содержат пары целых чисел, не превышающих по модулю  $10^8$  разделенных пробелами – координаты левого и правого концов окрашенного отрезка.

### Выходные данные

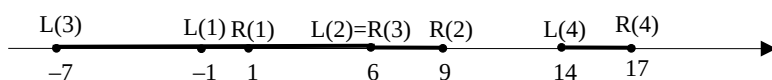
Выходной файл **task4.out** должен содержать целое число – длину самой большой образовавшейся окрашенной части прямой.

Ваша программа должна

- прочитать из файла **task4.in** исходное число отрезков  $n$  и координаты левых и правых концов окрашенных отрезков;
- подсчитать и вывести в файл **task4.out** длину самой большой образовавшейся окрашенной части прямой.

Пример:

task4.in	task4.out
4 -1 1 6 9 -7 6 14 17	16



Примечание. Для всех программ ограничение по времени прохождения одного теста — 1 секунда.

## Тесты и рекомендации по оценке 2015-16

### 12. Задание «Хоккей» – 100 баллов

Тест	task1.in	task1.out
1	2 2 1 5 6	0
2	4 1 0 5 2 4 4 2 10	1
3	6 1 1 2 2 3 4 5 3 7 6 0 1	1
4	6 1 1 2 2 3 4 5 3 7 6 0 1	2
5	10 7 2 5 4 0 5 8 6 1 1 4 4 4 3 3 9 2 9 1 5	2

За каждый тест по 20 баллов

### 13. Задание «Эксперимент» – 100 баллов

Тест	task2.in	task2.out
1	06:12:34 18:35:59	12:23:25
2	06:12:34 18:35:24	12:22:50
3	06:12:34 05:35:24	23:22:50
4	06:12:34 05:14:24	23:01:50
5	20:12:34 01:20:40	05:08:06

За каждый тест по 20 баллов

#### 14. Задание «Числа» – 100 баллов

Тест	task3.in	task3.out
1	4	16
2	10	288
3	21	57314
4	34	29860704
5	40	535828592

За каждый тест по 20 баллов

#### Задание 4. «Покраска» – 100 баллов

Тест	task4.in	task4.out
1	3 5 9 0 2 14 20	6
2	4 8 10 -6 -3 3 8 -9 -5	7
3	6 -2 0 4 7 5 13 15 25 -4 1 5 13	10
4	8 -18 -4 -18 -3 5 14 -10 -8 -1 4 1 4 0 -2 -2 0	15
5	20 565 955 656 970 461 854 549 802 88 286 24 227 512 907 509 831 74 306 94 279	Продолжение 40 69 41 65 42 83 70 122 485 881 762 894 649 710 487 725 359 950 32 43

За каждый тест по 20 баллов

Ограничение по времени прохождения одного теста — 1 секунда.

**2014-15**

**Текст муниципальной (районной/городской) олимпиады по информатике  
2014–2015 учебного года**

**15. «Конфеты» – 100 баллов**

В конце каждого урока физкультуры учитель проводит забег и дает победителю забега четыре конфеты, а всем остальным ученикам – по одной. К концу четверти Петя заслужил **a** – конфет, Ваня – **b**, а Толя – **c** (**a, b, c** ≤ 10<sup>9</sup>). Известно, что один из них пропустил ровно один урок физкультуры, участвуя в олимпиаде по математике, остальные же уроков не пропускали. Кто из детей пропустил урок?

Исходный файл **task1.in** содержит в единственной строке три натуральных числа, количества конфет, полученных мальчиками Петей, Ваней и Толей. Соседние числа разделены ровно одним пробелом. Выходной файл **task1.out** должен в первой строке содержать имя ученика, который пропустил урок.

Ваша программа должна

- прочитать из файла **task1.in** число конфет, полученных Петей, Ваней и Толей;
- найти и вывести в файл **task1.out** имя ученика, который пропустил урок.

Пример:

<b>task1.in</b>	<b>task1.out</b>
29 32 25	Толя

**16. Числа – 100 баллов**

Из соседних цифр натурального числа **n** (**n** < 10<sup>9</sup>) без изменения порядка их следования образуют число, не содержащее ведущих нулей. Сколько разных чисел (включая **n** и однозначные числа) при этом могут получиться?

Исходный файл **task2.in** содержит в единственной строке натуральное число **n**. Выходной файл **task2.out** должен содержать натуральное число – ответ.

Ваша программа должна

- прочитать из файла **task2.in** исходное число **n**;
- подсчитать и вывести в файл **task2.out** количество чисел, которые могут получиться.

Пример:

<b>task2.in</b>	<b>task2.out</b>
101	4

*Примечание.* Имеются в виду числа 101, 10, 0, 1.

**6. «Пять делителей» – 100 баллов.**

Найти сумму целых чисел из промежутка от 1 до **n** (**n** ≤ 2000000000), у каждого из которых ровно 5 делителей, включая 1 и само число.

Исходный файл **task3.in** содержит в единственной строке натуральное число **n**. Выходной файл **task3.out** должен содержать натуральное число – ответ, возможно очень длинное.

Ваша программа должна

- прочитать из файла **task3.in** исходное число **n**;
- подсчитать и вывести в файл **task3.out** сумму чисел из промежутка от 1 до **n**, которые имеют ровно 5 делителей.

Пример:

<b>task3.in</b>	<b>task3.out</b>
2014	722

**Продолжение на следующей странице**

**Текст муниципальной (районной/городской) олимпиады по информатике  
2014–2015 учебного года, продолжение**

**4. «Вирус» – 100 баллов**

Некоторые  $v$  ( $0 < v \leq 10$ ) клеток квадратного поля  $n \times n$  ( $n \leq 1000$ ) заражены вирусом. За каждый ход вирус заражает 4 соседние с ним клетки (от угловых клеток заражаются – 2 клетки, от прилежащих к границе поля – 3 клетки). Положение вирусов задано координатами  $(x, y)$  клеток на поле. Определите, за какое минимальное количество ходов будет заражено все поле.

Исходный файл **task4.in** содержит в первой строке два натуральных числа  $n$  и  $v$ , разделенных ровно одним пробелом. Каждая из последующих  $v$  строк содержит два разделенных пробелом числа – координаты  $x$  и  $y$  клетки поля, зараженной вирусом. Строки и столбцы нумеруются с единицы. Выходной файл **task4.out** должен содержать натуральное число – ответ.

Ваша программа должна

- прочитать из файла **task4.in** размер поля  $n$  и число вирусов  $v$ ;
- прочитать из файла **task4.in** координаты клеток  $x$  и  $y$ , зараженных вирусом;
- подсчитать и вывести в файл **task4.out**, за какое минимальное количество ходов будет заражено все поле.

Пример:

task4.in	task4.out
6 1 2 3	7

## Тесты и рекомендации по оценке 2014-15

### 17. Задание 1. «Конфеты» – 100 баллов

Тест	task1.in	task1.out
1	27 26 27	Ваня
2	35 48 33	Петя
3	50 48 30	Петя
4	31 34 27	Толя
5	35 37 32	Ваня

За каждый тест по 20 баллов

### 18. Задание 2. Числа – 100 баллов

Тест	task2.in	task2.out
1	114	5
2	2014	8
3	102003	12
4	123456789	45
5	10020233	18

За каждый тест по 20 баллов

### • Задание 3. «Пять делителей» – 100 баллов.

Тест	task3.in	task3.out
1	50	16
2	200	97
3	3000	3123
4	1500000	2170810
5	1999999999	161673334846

За каждый тест по 20 баллов

## Тесты и рекомендации по оценке 2014-15, продолжение

### Задание 4. «Вирус» – 100 баллов

Тест	task4.in	task4.out
1	7 1 4 2	8
2	25 2 2 21 20 3	27
3	250 5 1 1 125 110 250 250 3 222 250 1	137
4	499 8 50 250 100 200 100 300 50 150 50 350 299 250 199 325 400 287	385
5	1000 10 182 440 454 119 251 55 411 957 789 1 775 567 890 734 569 930 287 567 476 972	543

За каждый тест по 20 баллов

**Ограничение по времени прохождения одного теста — 1 секунда.**



**2013-14**

## Текст районной/городской олимпиады по информатике 2013–2014

### 1. «Многотомник» – 100 баллов

Многотомное собрание сочинений (не более 32 томов) стоит в произвольном порядке на полке. Разрешается брать любой том, и ставить его последним. Составить программу, которая определяет наименьшее число перестановок необходимое для того, чтобы упорядочить слева направо собрание сочинений, состоящее из  $n$  томов.

Ваша программа должна

запросить или прочитать из файла **order.in** число томов;

запросить или прочитать из того же файла начальный порядок томов;

найти и вывести на экран или в файл **order.out** наименьшее число перестановок, которое необходимо для упорядочения собрания сочинений.

Пример: Число томов 5 Начальный порядок томов 3 2 4 1 5

Минимальное число перестановок 4.

*Примечание.* Исходный текстовый файл **order.in** содержит в первой строке одно число  $n$  – количество томов в собрании. В второй строке  $n$  – чисел. Соседние числа разделены ровно одним пробелом. Выходной текстовый файл **order.out** должен в первой строке содержать единственное число – ответ

### 2. «Неравенства» – 100 баллов

Про числа  $a$ ,  $b$  и  $c$  известно, что среди них нет равных, а также результаты их попарного сравнения в виде трех непротиворечивых неравенств вида  $x < y$  или  $x > y$  в произвольном порядке. Необходимо записать соотношение между этими величинами в виде правильного двойного неравенства вида  $x < y < z$ .

Ваша программа должна

принять с клавиатуры или прочитать из файла **ineg.in** три результата попарного сравнения;

найти и вывести на экран или в файл **ineg.out** правильное двойное неравенство для величин  $a$ ,  $b$  и  $c$ .

Пример:

Первое неравенство	$b < c$
Второе неравенство	$a < c$
Третье неравенство	$a > b$
Ответ	$b < a < c$

*Примечание.* Исходный текстовый файл **ineg.in** содержит в каждой из первых трех строк один из результатов сравнения. Выходной текстовый файл **ineg.out** должен содержать двойное неравенство – ответ.

### 3. «К единице-2 – 100 баллов».

За один ход с числом делается такая операция: если число не делится на 3, то вычитаем 1, а если делится, то делим на 3. Так из числа 39 единица получается за 5 таких ходов (имеется в виду цепочка  $39 - 13 - 12 - 4 - 3 - 1$ ). Сколько существует натуральных чисел, каждое из которых превращается в единицу ровно за  $k$  ( $0 \leq k \leq 35$ ) таких ходов

Программа должна

Запросить число ходов  $k$  (натуральное число,  $0 \leq k \leq 35$ );

Найти и вывести на экран количество натуральных чисел, которые превращаются в единицу за  $k$  таких ходов.

Пример:

Число ходов	3	Всего натуральных чисел	6
-------------	---	-------------------------	---

*Примечание.* Для ввода и вывода информации можно использовать текстовые файлы **number.in** и **number.out** соответственно.

### 4. «Отрезки» – 100 баллов

На числовой оси заданы  $n$  ( $n < 100$ ) точек в порядке возрастания их координат (натуральные числа не превосходящие 1000). Соседние точки соединяются отрезками, но так, чтобы ни одна из точек не оказалась принадлежащей сразу двум отрезкам. При этом сумма длин проведенных отрезков должна быть максимальной.

Пример: Число точек: 6 Координаты точек: 1 3 5 12 15 16

Максимальная длина отрезков 10 Всего отрезков 4

Выбранные отрезки: 1-2 3-4 5-6 Всего отрезков — 3

Исходные данные должны вводиться с клавиатуры или читаться из текстового файла **section.in**, а найденные результаты выводиться на экран или в текстовый файл **section.out**. Исходный файл **section.in** содержит в первой строке одно число  $n$  – количество точек. Во второй строке  $n$  чисел – координаты точек. Соседние числа разделены ровно одним пробелом. Выходной текстовый файл **section.out** должен в первой строке содержать единственное число – максимальную длину отрезков, во второй строке – число выбранных отрезков, а в последующих – по два числа, задающих номера точек-концов очередного выбранного отрезка, разделенных тире (без пробелов).

## Тесты и рекомендации по оценке 2013-14

### Задание 1. «Многотомник» – 100 баллов

Тест	Исходные данные (файл <b>order.in</b> )	Ответ
1	5 5 4 3 2 1	4
2	7 1 7 2 6 3 5 4	3
3	10 8 9 10 1 2 3 4 5 6 7	3
4	15 1 12 2 15 13 14 3 4 11 10 5 6 8 9 7	8
5	20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20	0

За каждый тест по 20 баллов

### Задание 2. «Неравенства » – 100 баллов

Тест	Исходные данные (файл <b>ineg.in</b> )	Ответ
1	c>b a<c b>a	a<b<c
2	a<b a<c c<b	a<c<b
3	c>b a>c a>b	b<c<a
4	a<b a>c c<b	c<a<b
5	a>c b<a c<b	c<b<a

За каждый тест по 20 баллов

### Задание 3. «К единице-2» – 100 баллов

Тест	Число ходов (файл <b>number.in</b> )	Количество чисел
1	5	20
2	10	423
3	12	1431
4	30	830047505
5	35	1748130326

Оценка: 20 баллов за тест.

**Задание 4. «Отрезки» – 100 баллов**

Тест	Число точек и их координаты (файл <b>section.in</b> )	Ответ
1	3 1 2 4	2 1 2-3
2	4 1 3 10 14	7 1 2-3
3	7 1 11 19 20 22 25 36	23 3 1-2 4-5 6-7
4	10 10 12 29 37 38 39 40 44 48 49	23 4 2-3 4-5 6-7 8-9
5	19 1 4 24 26 27 57 60 62 102 103 108 148 152 155 185 187 191 213 216	182 6 2-3 5-6 8-9 11-12 14-15 17-18

За каждый тест по 20 баллов

Ограничение по времени прохождения одного теста — 1 секунда.

**2012-13**

**Текст районной/городской олимпиады по информатике  
2012–2013 учебного года**

**1. «Сделал дело – гуляй смело» – 100 баллов**

Вовочка и Марья Ивановна (школьная учительница Вовочки) должны проверить 30 школьных заданий. Учительница не отпустит его играть с папой в футбол прежде, чем закончится проверка всех заданий – ее и Вовочкиных. Папа ждет Вовочку с нетерпением, и уже разминается на футбольном поле. Как Вовочке и учительнице лучше распределить между собой задания, чтобы Вовочка смог пораньше освободиться? На проверку одного задания он тратит в среднем  $m$  минут, а Марья Ивановна –  $k$  минут. Найдите наименьшее время (в минутах), которое им необходимо будет потратить на проверку всех заданий.

Ваша программа должна

- запросить  $m$  и  $k$ ;
- найти и сообщить наименьшее время, которое необходимо будет потратить на проверку всех заданий.

Пример:

Исходные данные	17	5
Наименьшее время	119	

**2. «Прогрессия» – 100 баллов**

От записанной на доске арифметической прогрессии, состоящей из натуральных чисел, остались только первый член  $a$ , последний член  $c$  и еще один член  $b$ .

Ваша программа должна

- запросить значения  $a$ ,  $b$  и  $c$  (натуральные числа,  $0 < a < b < c < 2^{31}-1$ );
- найти и сообщить разность этой прогрессии. Если таких прогрессий несколько, то той, которая имеет меньше всего членов.

Пример:

Исходные данные	3	23	113
Ответ	10		

**3. «Кенгуру» – 100 баллов**

Суперкенгуру может прыгать по прямой вправо и влево и совершать гигантские прыжки. Длина его первого прыжка составляет 1 м, второго — 2 м, третьего — 3 м и так далее (длина каждого прыжка всегда на 1 метр больше, чем предыдущего). Через какое минимальное количество прыжков суперкенгуру окажется на расстоянии  $d$  метров от исходной точки  $O$ ?

Программа должна

- запросить расстояние  $d$  (натуральное число,  $0 < d < 2^{28}-2$ );
- найти и вывести на экран минимальное количество прыжков для попадания в точку на расстоянии  $d$ .

Примеры:

Расстояние	17
Минимальное число шагов	6

*Примечание.* Имеется в виду последовательность прыжков  $1-2+3+4+5+6$ , дающая расстояние 17.

**4. «Число» – 100 баллов**

Вася Пупкин из цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 пытается составить число  $a_1a_2...a_n$  такое, в котором

1. Первая цифра не ноль;
2. Нет повторяющихся цифр;
3. Число  $a_1a_2$  делится на 2,  $a_1a_2a_3$  делится на 3, ...,  $a_1a_2...a_n$  делится на  $n$ .

Помогите ему. Составьте программу, которая находит наибольшее число, начинающееся на введенную пользователем по запросу первую цифру  $a_1$  и удовлетворяющую поставленным Васей условиям.

Например:

Первая цифра	5
Искомое число	5612047

*Примечание.* Для всех программ ограничение по времени прохождения одного теста — 1 секунда.

## Тесты и рекомендации по оценке

### 1. Задание 1. «Сделал дело – гуляй смело» – 100 баллов

Тест	Исходные данные	Ответ
1	8 3	66
2	8 4	80
3	31 1	30
4	2 65	60
5	10 7	126

За каждый тест по 20 баллов

### Задание 2. «Прогрессия» – 100 баллов

Тест	Исходные данные	Ответ
1	14 56 60	2
2	2012 3001 101901	989
3	14 5014 10014	5000
4	323 1208 197324	177
5	123456 1339671 1474806	135135

За каждый тест по 20 баллов

### Задание 3. «Кенгуру» – 100 баллов

Тест	Исходные данные	Ответ
1	18	7
2	21	6
3	3000	79
4	2025078	2012
5	300000000	24495

За каждый тест по 20 баллов

### Задание 4. «Число» – 100 баллов

Тест	Исходные данные	Ответ
1	4	48965
2	6	6812043
3	1	12965408
4	8	801654723
5	3	3816547290

За каждый тест по 20 баллов

Рекомендуется для всех программ использовать ограничение по времени прохождения одного теста — 1 секунда.

Справки по телефону 417–76–49, E-mail a-g-b@mail.ru (Баханский А.Г.)

**2011-12**



## Текст районной/городской олимпиады по информатике 2011–2012 учебного года

### 1. «Акция» – 100 баллов

ЗАО «Титан» выпустила первую опытную партию новых катеров. В рамках рекламной компании решено организовать водно-моторный переход Н.Новгород-Астрахань. В походе запланировано участие  $n$  ( $n \leq 100$ ) катеров. На место сбора прибыли катера, на которых находились соответственно  $a_1, a_2, \dots, a_n$  участников похода ( $a_i \leq 32$ ).

Для обеспечения максимально возможной равномерности загрузки катеров было решено перераспределить часть участников, но так чтобы число пересадок было минимально.

Ваша программа должна по введенным исходным данным сообщать единственное число, минимально возможное количество пересадок, обеспечивающих максимально равномерную загрузку катеров.

Например:

Число катеров 3

Число участников похода по катерам 1, 2, 4

Минимальное число пересадок 1 (соответствующее распределение 2, 2, 3).

### 2. «Квадрат» – 100 баллов

На координатной плоскости задан квадрат. Одна из его диагоналей соединяет точки  $(0, 0)$  и  $(1000, 1000)$ . На периметре квадрата заданы целочисленными координатами две точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$ . Найти длину наименьшего пути от точки  $A$  до  $B$ , если перемещение возможно только по периметру.

Ваша программа должна по введенным исходным данным сообщать единственное число, длину минимального пути от  $A$  до  $B$  по периметру.

Пример:

Координаты точки A 300, 0

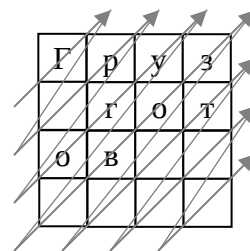
Координаты точки B 500, 1000

Длина наименьшего пути 1800

### 3. «Декодирование» – 100 баллов

Агент 0013 (позывной Вася) придумал такой способ кодирования пересылаемых сообщений:

1. Определяется длина пересылаемого сообщения (например, для сообщения «Груз готов» это 10).
2. Заготавливается квадратная таблица минимального размера, но содержащая ячеек не менее чем длина сообщения (в примере это  $4 \times 4$ ).
3. Ячейки таблицы последовательно заполняются символами сообщения слева направо сверху вниз, а оставшиеся пустыми – пробелами (смотри рисунок).
4. Кодированное сообщение читается по направлению параллельному вспомогательной диагонали в соответствии со стрелками на рисунке. Возможные последние незначащие пробелы игнорируются. Таким образом, кодированное сообщение будет «Г\_рогу\_воз\_т» (для предупреждения ошибок здесь пробелы обозначены символом «\_»).



Вам необходимо составить программу раскодирования сообщений агента, содержащих до 127 символов.

Еще один пример:

Кодированная строка

**Пцрсеооср**

Сообщение

**Процессор**

### 4. «Делители» – 100 баллов

Составьте программу, которая по натуральному  $n$  ( $n \leq 100$ ) находит и сообщает число различных натуральных делителей (включая 1 и само число) числа  $n!$  ( $n! = 1 \times 2 \times 3 \times \dots \times n$ ).

Например:

Число  $n$  8

Число делителей 96

**Примечание.** Для всех программ ограничение по времени прохождения одного теста — 1 секунда.

## Тесты и рекомендации по оценке 2011-12

### Задание 1. «Акция» – 100 баллов

Тест	Исходные данные	Пример правильного ответа
1	5 1, 2, 3, 4, 5	3
2	6 5, 6, 6, 5, 6, 5	0
3	7 1, 8, 2, 7, 3, 6, 4	6
4	8 32, 31, 30, 1, 2, 3, 24, 25	48
5	10 1, 10, 20, 30, 25, 15, 5, 2, 3, 18	43

За каждый тест по 20 баллов

### Задание 2. «Квадрат» – 100 баллов

Тест	Исходные данные	Ответ
1	(666, 1000), (1000, 1000)	334
2	(0, 255), (700, 1000)	1 445
3	(1000, 100), (10, 1000)	1 890
4	(0, 150), (1000, 950)	1 900
5	(200, 0), (850, 1000)	1 950

За каждый тест по 20 баллов

### Задание 3. «Декодирование» – 100 баллов

Тест	Исходные данные	Ответ
1	Ирнимф_као_ат	Информатика
2	Оилпаифодмто_апири_кмнеа	Олимпиада по информатике
3	Влсооег_ж_лвес_у_н_у	Все сложено в углу
4	Пмрииооергар_ортма-ва_ья_ам_т_вн_нгт_ор_с	Программирование – вторая грамотность
5	Жпд2еуб4х_1р_оуа1янвс_о_г2я_о0_д	Жду успехов 24 ноября 2011 года

За каждый тест по 20 баллов

### Задание 4. «Делители» – 100 баллов

Тест	Исходные данные	Ответ
1	6	30
2	9	160
3	16	5 376
4	32	5 529 600
5	100	39 001 250 856 960 000

За каждый тест по 20 баллов

Рекомендуется для всех программ использовать ограничение по времени прохождения одного теста — 1 секунд.



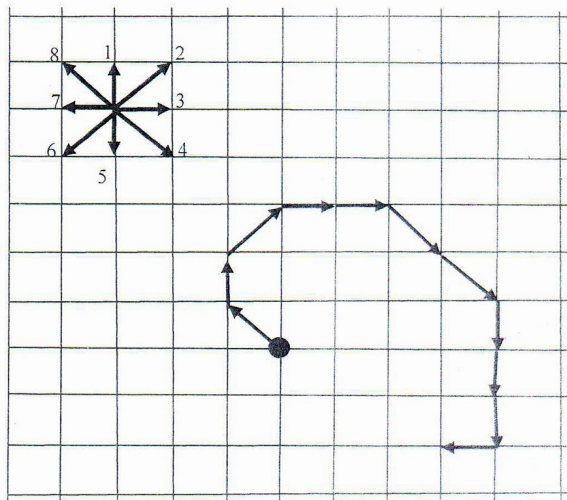
**Текст районной/городской олимпиады по информатике  
2009–2010 учебного года**

**1. «Маршрут» (25 баллов)**

Строка, содержащая символы «1», «2», «3», «4», «5», «6», «7», «8», является описанием маршрута передвижения фишки по узлам решетки со стороной равной 1. Переход осуществляется в соседний узел, заданный символом (см. рисунок). Приведенный на рисунке маршрут задается строкой «81233445557».

Составить программу, которая по маршруту, заданному строкой, вычисляет и сообщает максимальное отклонение фишки от начальной точки. Результат должен быть выведен с точностью до трех знаков после запятой.

Например, для приведенного на рисунке маршрута ответ 4,472.



**2. «Число» (25 баллов).**

Для заданного натурального числа  $N$  ( $N \leq 32000$ ) найти наименьшее натуральное число  $K$  такое, что  $K^N$  делится на  $N$ .

Например, если  $N=24$ , то  $K=6$ .

**3. «Билеты» (25 баллов)**

Написать программу определения количества  $2*N$ -значных билетов, у которых сумма первых  $N$  десятичных цифр равна сумме  $N$  последних десятичных цифр; при этом  $N \leq 10$  - натуральное число.

Например, если  $N=2$ , то счастливых билетов  $K=670$ .

**4. «Монеты» (25 баллов)**

На столе разложено  $N$  монет. Из них  $K$  монет лежат гербом вниз (решка), а остальные вверх (орел). За один ход разрешается одновременно перевернуть любые 7 монет. Окончательная цель повернуть монеты так, чтобы они все лежали гербом вниз за наименьшее количество ходов.

Составить программу, которая

- Запрашивает число монет  $N$  ( $7 \leq N \leq 100$ );
- Запрашивает число монет лежащих гербом вниз  $K$  ( $0 \leq K < N$ );
- Находит и сообщает наименьшее число ходов, за которое можно перевернуть все монеты гербом вниз или сообщить что это невозможно.

Например, если  $N=13$  и перевернута 1-я монета ( $K=1$ ), то это достигается за 2 хода:

ход 1 – переворачиваем монеты с номерами 1, 8–13;

ход 2 – переворачиваем монеты с номерами 1–7.