

**Фестиваль «Юные интеллектуалы Среднего Урала»**  
**Муниципальный этап Всероссийской олимпиады по информатике**  
**2010–2011 учебный год**  
**9 класс**

*Время выполнения задач — 4 часа*  
*Ограничение по времени — 2 секунды на тест*  
*Ограничение по памяти — 64 мегабайта*

*Уважаемый участник!*

В настоящее время решения на олимпиадах по информатике проверяются автоматически. Ваша задача — написать программу, которая по заданным входным данным вычисляет и выводит выходные данные. Когда вы сдаёте решение на проверку, проверяющая программа «подсовывает» вашему решению тестовые наборы входных данных, запускает вашу программу и затем анализирует выданный ею результат: если в задаче возможно несколько правильных ответов, то программа может выводить любой из них, если в условии не указано иное.

Ваше решение должно читать входные данные из файла **input.txt** в описанном формате, решать задачу, и выводить результат в файл **output.txt**. Всякая строка в файле **input.txt** завершается переводом строки. Программа должна всегда завершаться с кодом 0 (иначе тестирующая программа считает, что в ходе работы произошла ошибка) — то есть, командой **halt(0)**; или просто достижением конца текста, если Вы пишете на Паскале, или **return 0**; для программ на C/C++.

Считывать информацию из текстового файла так же просто, как и с клавиатуры — в Бейсике, Паскале и C/C++ для ввода с клавиатуры и из файла используются или одни и те же операторы или же операторы со сходной структурой вызова. То же касается и вывода в файл. Возникают лишь небольшие отличия, связанные с файловыми переменными (см. памятку по работе с файлами).

Ваша программа не должна ничего выводить на экран, а также ждать какого-либо ввода пользователя. Распространённой ошибкой является ситуация, когда после окончания работы программа ждёт нажатия какой-либо клавиши. Этого быть не должно.

Те, кто программирует на Паскале в среде Borland Pascal, не должны использовать модуль CRT: программа не должна содержать команды **uses crt**;

Программа должна выдавать в выходной файл ту и только ту информацию, которая описана в формате вывода. Более того, вывод программы должен в точности удовлетворять формату, описанному в условии задачи (заглавные/строчные буквы, наличие/отсутствие пробелов, переводы строк). В противном случае, программа считается нерабочей и оценивается в 0 баллов.

Все задачи считаются равноценными и имеют одинаковую максимальную оценку в 100 баллов. Все тесты в рамках каждой задачи также равноценны.

**9.1. «Кратность делителя».** На парте в кабинете математики Петя Торопыжкин обнаружил натуральные числа  $n$  и  $m$ , причём  $m$  больше единицы. Его заинтересовало, является ли число  $m$  делителем числа  $n$  и, если да, то какой кратности. Помогите ему — напишите программу, отвечающую на этот вопрос.

**Формат ввода:** В единственной строке записаны целые числа  $n$  и  $m$ , разделённые пробелами ( $1 \leq n \leq 10^9$ ;  $2 \leq m \leq 10^9$ ).

**Формат вывода:** Выведите единственное неотрицательное целое число — кратность делителя  $m$  у числа  $n$ . Если  $m$  не является делителем  $n$ , следует вывести 0.

Пример 1		Пример 2	
input.txt:	output.txt:	input.txt:	output.txt:
120 2	3	1034 3	0

**9.2. «Кольцо из проводов».** У Пети Торопыжкина есть набор проводов, концы которых обжаты в разъёмы двух типов. Существует три типа проводов: 1–1 — их у Пети  $n$  штук, 2–2 — их  $m$  штук и 1–2 — их  $k$  штук (указаны типы разъёмов на концах). При этом два разъёма можно соединять друг с другом только если они одного типа. Напишите программу, которая сможет по заданным  $n$ ,  $m$  и  $k$  определить, какое минимальное количество проводов нужно исключить из набора, чтобы из оставшихся можно было собрать кольцо.

**Формат ввода:** В единственной строке записаны целые числа  $n$ ,  $m$  и  $k$  — количество 1–1, 2–2 и 1–2 проводов, соответственно ( $0 \leq n, m, k \leq 10^9$ ). Числа разделены пробелами.

**Формат вывода:** Выведите единственное целое число — минимальное количество проводов из набора, которые нужно выкинуть, чтобы из оставшихся можно было составить кольцо. Если можно собрать кольцо, ничего не выкидывая, выведите 0. Если никакое кольцо составить в принципе невозможно, выведите -1.

Пример 1		Пример 2	
input.txt:	output.txt:	input.txt:	output.txt:
10 5 2	0	7 0 5	1

**9.3. «Сколько приложений?».** На смартфоне Пети Торопыжкина установлено много полезных приложений, которыми он часто пользуется. Однако Петя сам никогда не закрывает приложения. Если необходимость в запущенном приложении отпадает, он отправляет его в фоновый режим, не выгружая из памяти. Все приложения устроены таким образом, что позволяют запустить одновременно лишь одну свою копию, так что если Петя запустит приложение, которое уже было запущено ранее, то оно просто выйдет из фонового режима и второго запущенного его экземпляра не возникнет.

Напишите программу, которая по списку названий запущенных Петей приложений (в хронологическом порядке) определит, сколько сейчас приложений в памяти (в активном или фоновом режиме).

**Формат ввода:** В первой строке записано целое число  $n$  — сколько раз Петя запускал приложения ( $1 \leq n \leq 5\,000$ ). В следующих  $n$  строках перечислены названия приложений в том порядке, в котором их запускал Петя. Название приложения — строка, состоящая из заглавных и строчных латинских букв, длиной не более 20 символов. Приложения, названия которых отличаются только регистром букв, следует считать различными.

**Формат вывода:** Выведите целое число — количество приложений, находящихся в памяти смартфона после всех манипуляций Пети.

#### Пример 1

```
input.txt:  output.txt:
1           1
NotePad
```

#### Пример 2

```
input.txt:  output.txt:
7           4
NotePad
Word
NotePad
CorelDraw
Word
AdobePhotoshop
Word
```

**9.4. «Питание насекомых».** У Пети Торопыжкина живёт ручной жучок Ипполит. Уходя в школу, Петя оставляет Ипполиту на письменном столе квадратную плитку шоколада размера  $2 \times 2$ . Центр плитки совпадает с началом декартовой системы координат, которую Петя ввёл на столе, а стороны плитки параллельны осям. Однако Ипполит нетерпелив: он падает с потолка куда придётся. Если он не попадает на плитку, то бежит к ближайшей её точке, где и принимается за трапезу. Напишите программу, которая, зная точку падения Ипполита, находит ближайшую к нему точку шоколадной плитки.

**Формат ввода:** В единственной строке записаны два вещественных числа, по модулю не превышающие 100, с не более чем двумя знаками после десятичной точки — координаты точки, в которую упал Ипполит. Числа разделены пробелом.

**Формат вывода:** Выведите два вещественных числа, разделённые пробелом — округлённые до сотых координаты точки, в которую следует бежать Ипполиту.

#### Пример 1

```
input.txt:  output.txt:
1.10 1.2    1.00 1.00
```

#### Пример 2

```
input.txt:  output.txt:
0.9 1.1     0.90 1.00
```

**9.5. «Математический бильярд».** На координатной плоскости в первом квадранте расположен прямоугольный бильярдный стол так, что его стороны парал-

лельны координатным осям, а один из углов находится в начале координат. Размеры стола  $a \times b$  ( $a$  — вдоль оси  $Ox$ ,  $b$  — вдоль оси  $Oy$ ). Точечный бильярдный шар расположен в точке  $(x, y)$ , находящейся на поверхности стола, и движется со скоростью  $v = (v_x, v_y)$ . Шар движется без трения (сохраняя величину скорости) и отражается от бортов по закону «угол падения равен углу отражения». Если шар попадает точно в угол, считается, что он мгновенно испытывает два отражения и, стало быть, изменяет вектор скорости на противоположный. Все линейные величины заданы в метрах, скорости — в метрах в секунду. Вопрос: в какой точке шар окажется через  $t$  секунд?

**Формат ввода:** В первой строке записаны числа  $a$  и  $b$ , во второй — числа  $x$  и  $y$ , в третьей — числа  $v_x$  и  $v_y$ . Числа разделены пробелом. В четвёртой строке записано единственное число  $t$ . Все числа целые, неотрицательные и не превосходят 10 000.  $a$  и  $b$  строго положительные.

**Формат вывода:** Выведите два неотрицательных целых числа, разделённые пробелами — координаты шара через  $t$  секунд.

#### Пример

```
input.txt:  output.txt:
10 10      8 5
1 1
1 2
7
```