

**Фестиваль «Юные интеллектуалы Среднего Урала»**  
**Муниципальный этап Всероссийской олимпиады по информатике**  
**2010–2011 учебный год**  
**8 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.

**8.1. «Сажаем картошку».** Весной Петя Торопыжкин помогал бабушке в деревне сажать картошку. Картофельное поле состоит из  $n$  рядов, в каждом из которых выкопано  $m$  лунок. В одну лунку нужно посадить одну картофелину. В один мешок входит  $k$  картофелин. Сколько мешков понадобится для засева бабушкиного поля? (Мешки можно покупать только целиком!)

**Формат ввода:** В единственной строке записаны целые числа  $n$ ,  $m$  и  $k$ , разделённые пробелами ( $1 \leq n, m \leq 150$ ;  $1 \leq k \leq 10\,000$ ).

**Формат вывода:** Выведите единственное целое число — количество мешков картошки, которое надо купить.

**Пример**

```
input.txt:   output.txt:
1 1 2        1
```

**8.2. «Самый большой монитор».** На день рождения начинающему программисту Пете Торопыжкину родители решили подарить новый монитор. Они знают, что с точки зрения Пети монитор тем лучше, чем больше на нем пикселей. Напишите программу, которая из заданного набора мониторов (для каждого монитора известны его ширина и высота в пикселях) выберет наилучший.

**Формат ввода:** В первой строке записано целое число  $n$  — количество типов мониторов ( $1 \leq n \leq 150$ ). В  $k$ -й из следующих  $n$  строк записаны два целых числа  $w_k$  и  $h_k$  — ширина и высота монитора  $k$ -го типа в пикселях, разделённые пробелом ( $1 \leq w_k, h_k \leq 150$ ).

**Формат вывода:** Выведите единственное целое число — номер наилучшего монитора (мониторы занумерованы в порядке, в котором они перечислены в исходном списке). Если есть несколько мониторов с наибольшим количеством пикселей, выведите номер первого из них.

**Пример**

```
input.txt:   output.txt:
3            2
1 1
5 10
12 2
```

**8.3. «На уроке английского».** В тетради по английскому языку Петя Торопыжкин написал длинное слово, состоящее только из заглавных букв латинского алфавита. Ему стало интересно, сколько различных букв латинского алфавита содержит это слово. Помогите ему — напишите программу, которая по введённой строке будет выдавать эту информацию.

**Формат ввода:** Единственная строка содержит слово, которое написал Петя. Это слово состоит только из заглавных латинских букв, непусто и имеет длину не более 200 символов. Слово завершается переводом строки.

**Формат вывода:** Выведите единственное целое число — количество различных букв в слове, которое написал Петя.

**Пример**

```
input.txt:  output.txt:
ABCDEA      4
```

**8.4. «Сложение дробей».** Успешно справившись с домашним заданием по английскому языку и сев за компьютер с новым монитором, Петя Торопыжкин занялся на компьютере математикой. Однако результат первого же действия сильно удивил его: он разделил 10 на 3 и вместо ожидаемого  $3\frac{1}{3}$  или хотя бы  $\frac{10}{3}$  увидел на экране 3.3333333334. Чтобы исправить этот «недостаток» компьютера, Петя решил научить компьютер работать с обыкновенными дробями, а для начала — складывать их. Помогите Пете — напишите программу, которая складывает две положительные обыкновенные дроби и выдаёт результат в виде обыкновенной несократимой дроби (возможно, неправильной). Дроби задаются числителем и знаменателем.

**Формат ввода:** В первой строке записаны целые числа  $p_1$  и  $q_1$  — числитель и знаменатель первой дроби, во второй — целые числа  $p_2$  и  $q_2$  — числитель и знаменатель второй дроби;  $1 \leq p_1, q_1, p_2, q_2 \leq 10\,000$ . Числа разделены пробелами.

**Формат вывода:** Выведите два целых числа, разделённые пробелом: числитель и знаменатель несократимой дроби, являющейся суммой заданных дробей. Если в результате получилось целое число, следует выдать знаменатель 1.

Пример 1	Пример 2	Пример 3
input.txt:  output.txt: 5 3          13 6 1 2	input.txt:  output.txt: 3 4          11 4 2 1	input.txt:  output.txt: 5 4          2 1 3 4

**8.5. «Сортируем числа».** Дана последовательность, состоящая из чисел 1, 2 и 3. Длина последовательности не превосходит 200. Можно брать любые два элемента и менять их местами. Требуется за наименьшее число таких перестановок поставить на первые места в последовательности все единички, дальше все двойки и потом все тройки.

**Формат ввода:** В первой строке записано целое число  $n$  ( $1 \leq n \leq 200$ ). В следующей строке перечислены  $n$  чисел, каждое из которых равно 1, 2 или 3. Числа разделены пробелами.

**Формат вывода:** Выведите неотрицательное целое число — наименьшее количество перестановок, за которое можно отсортировать последовательность.

**Пример**

```
input.txt:  output.txt:
5           3
2 3 3 1 1
```