

Некоторые тесты

Данные тесты не являются официальными. Жюри использовало для проверки решений наборы тестов, отличные от приведенных ниже.

Если Вы уже прочитали условие задачи, написали решение, и никак не можете найти в нем маленькую ошибочку, то эти тесты, вероятно, могут Вам помочь.

Основные идеи и методы решений задач будут опубликованы позже

Задача А. «Круглая цифра»

Входной файл input.txt	Выходной файл output.txt
2	0 2
16	0 2
160	0 16
17	1 3
93313	4991 8

Задача В. «Бразды»

Входной файл input.txt	Выходной файл output.txt
1 10000 10000 1000	NO
5 5 1 2 3 4 5 1 2 3 4 5	9 3
5 13 4 4 4 4 4 1 1 1 1 1	6 5
5 15 4 4 4 4 4 1 1 1 1 1	6 5
5 9 1 2 3 4 5 1 2 3 4 5	30 4
1 9999 10000 1000	49995000000 1

Задача С. «Восточные сладости»

Входной файл input.txt	Выходной файл output.txt
10000 109999 109998 109997	9999 9998 9997

Заметим, что, пожалуй, единственная возможность ошибиться в этой задаче – неверно выбрать тип данных

Задача D. «Точное время»

Входной файл input.txt	Выходной файл output.txt
00:01 01:01 01:02 1000 10 10	51 00:32

00:01 00:03 00:03 100 10 10	6 00:02
00:01 00:03 00:03 100 10 100	NO
00:01 00:03 00:03 100 10 99	1 00:03
00:01 00:03 00:03 100 1 1	51 00:02
00:01 00:03 00:08 100 1 1	NO

На что следует обратить внимание при решении и отладке.

Во-первых, выходной файл должен совпадать посимвольно (пропускать нули при записи времени недопустимо).

Во-вторых, следует проверить «крайние случаи» (тесты 3 и 5 в приведенной выше таблице).

В-третьих, использование вещественных, а не целых типов данных требует более аккуратного программирования.

Задача Е. «Путь»

По мнению жюри, это самая сложная задача в соревновании, и для участников было бы полезно придумать набор тестов самостоятельно.

Задача F. «Сон купца Хасана»

Данная задача является «симулятором» – т.е. требует точного последовательного воспроизведения действий, описанных в условии. Единственное, что нужно тщательно контролировать, – это точность вычислений (где возможно, крайне желательно использовать целые типы данных)

Задача G. «Манускрипт»

Приведенные ниже тесты некорректны с точки зрения условия задачи.

Однако, во-первых, правильно написанная программа, скорее всего, должна корректно работать и на таких тестах, а, во-вторых (в силу простоты), эти тесты могут оказать помощь в отладке.

Входной файл input.txt	Выходной файл output.txt
9 0 0 0 0 0 0 0 5 10	0 21 6 21 3
Пояснение: $0\ 5\ 10\ (+6) = 21$ $0\ 0\ 0\ 0\ 0\ 0\ (+21) = 21$	

9 0 0 0 0 0 0 0 5 10	0 20 5 20 4
Пояснение (сравните с предыдущим тестом): 0 0 0 10 (+10) = 20 0 0 0 0 5 (+15) = 20	
9 0 0 0 0 0 0 0 0 40	5 41 1 36 8

Задача Н. «Желания»

Представляется, что в данной задаче довольно сложно ошибиться в реализации, если идея решения верна. Приведем один тест, отличающийся от имеющегося в тексте задачи.

Входной файл input.txt	Выходной файл output.txt
10 6 9 0 9 8 0 10 7 0 9 0 10 9 0 8 10 7 0 8 0 10 0 8 10 6 0 9 0 4 3 5 0 5 4 3 1 0 1 0 5 3 2 1 0 4 3 0 3 1 5 0 2 4 3 5 0 3 2 5 4 0 1 3 0 2 1 4 0	3 3 3 4 3 2 4 2 3 3

Задача I. «Смесь»

Вместо многоточия в последнем тесте, конечно, должны быть подставлены все числа от 6 до 29999

Входной файл input.txt	Выходной файл output.txt
3 3 2 1	5.91030
5 1 1 1 1 1	4.8808980000
15 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	14.4186431400
5 1 10 100 1000 10000	10987.6964561100
4 2300 2400 2600 2700	9801.00000000
30000 1 2 3 4 5 [...] 30000	388540681.6181762800 (C++) 388540681.618176377 (Delphi)

Задача J. «Послание»

Кроме теста, приведенного в условии, добавим еще один

Входной файл input.txt	Выходной файл output.txt
18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 6 if you think about.	if youthink about.

Если Ваше решение проходит эти тесты (представленный здесь и в условии), то, скорее всего, проблема может состоять в не совсем удачном использовании функций обработки строк. В частности, если Вы написали решение на Delphi, попробуйте сравнить быстродействие программы при использовании `copy(s, i, 1)` и `s[i]` для обращения к символу строки, проанализируйте, какой тип для строки Вы используете. Если же Ваше решение написано на C++, то известной сложностью является то, что метод `getline()` компилятора g++ «дочитывает» символ `0xD` (`'\r'`) стандартного перевода строки `"\r\n"`.