Система оценивания решений задач и
комплекты тестов проверки решений задач

Каждая задача оценивается баллами, указанными в текстах заданий. Результатом решения задачи является программа или алгоритм, сохраненный в среде программирования.

Все представленные на проверку решения участников сначала должны проходить предварительное тестирование на тестах из примера или примеров, приведенных в условии задачи. Если на этих тестах решение участника выдает правильный ответ, то тогда это решение принимается жюри на окончательную проверку, которая после завершения тура осуществляется на всех тестах из заданного набора тестов для этой задачи. В противном случае, решение участника считается неверным, и за него участнику не начисляются какие-либо баллы.

Для олимпиады даются 4 задачи: 2 простых, 1 средней сложности и 1 сложная. В следующей таблице дана сравнительная характеристика всех 4 задач

Задание	**Тематика**	**Сложность алгоритма**	**Сложность реализации**	**Баллы**
1	Операции с числами	Простая	Простая	100
2	Массивы	Простая	Средняя	100
3	Математика, жадный алгоритм	Средняя	Средняя	100
4	Перебор	Сложная	Средняя	100

Задача №1. Деление нацело.

Задачу можно решать либо в строчках (преобразовав исходное число в строку, и поочередно меняя последние символы, преобразовывать строку обратно в число и проверять делимость), либо в числах (используя mod и div).

При решении вторым способом – вначале заменяем последние 2 цифры числа N на 00 (поделив N нацело на 100, и умножив результат на 100). Дальше в цикле от 0 до 99, получаем поочередно все возможные числа с замененными последними 2 цифрами (каждое такое число – это [n/100]*100+i, где i – переменная цикла), и проверяем каждое такое число на делимость на k. Первое же встреченное число, которое делится на k – выводим, и прерываем цикл.

Решение на задачу будет существовать всегда, потому что k находится в диапазоне от 1 до 100, и значит среди 100 подряд идущих чисел хотя бы одно всегда будет делиться на k.

Задача №2. Много квадратов.

Для решения задачи подсчитаем для каждой длины палочки - сколько таких палочек есть у Васи. Поскольку длины палочек - целые числа от 1 до 1000, можно это считать в массиве от 1 до 1000, увеличивая массив от длины палочки на 1 для каждого числа из входных данных.

Затем для каждой длины палочки можно определить, сколько квадратов можно составить из таких палочек (кол-во палочек делить на 4). Сложив количество квадратов для каждой возможной длины от 1 до 1000 получим ответ.

Задача №3. Сломанные роботы

Рассмотрим команду "L". Возможны 4 варианта ее выполнения 2 роботами:

- Оба робота выполняют ее. Относительное расстояние между роботами не меняется.
- Оба робота не выполняют ее. Относительное расстояние между ними также не меняется.
- 1-й робот выполняет, 2-й - нет. В этом случае 1-й робот сдвинется влево относительно 2-го робота.
- 1-й робот не выполняет, 2-й - выполняет. В этом случае 2-й робот сдвинется влево относительно 1-го робота.

Переформулируя, возможны следующие варианты:

- Расстояние по горизонтали между роботами не меняется - используем этот вариант, если координата *x* у роботов уже одинаковая.
- Расстояние по горизонтали между роботами увеличивается - для решения задачи этот вариант не нужен.
- Расстояние по горизонтали между роботами уменьшается - используем этот вариант, если координаты *x* роботов разные.

Аналогично такие же варианты возможны для всех команд (для U/D - тоже самое для координаты *y*).

Таким образом, для решения задачи просто идем по строке слева направо и для каждой команды выбираем нужный нам вариант (или не меняем расстояние между роботами по горизонтали/вертикали - игнорируем команду обоими роботами, или сокращаем расстояние на 1).

Если в какой-то момент времени координаты роботов сравнялись, то выводим YES и полученные команды каждого робота.

Если дойдя до конца строки команд координаты роботов все еще разные, значит они не могут встретиться (выводим NO).

Задача №4. Язык программирования XY.

При заданных ограничениях пройдут различные варианты решения задачи, в том числе и обычный перебор всех возможных строк из символов X,Y (при N=1000 длина строки 16 символов, т.е. всего различных строк такой длины 65536). Далее же предлагается оптимальное и быстрое решение этой задачи, даже при значительно больших ограничениях на N.

Заметим следующие свойства операций языка программирования XY:

1) Если X>Y, то последней была операция X (X:=X+Y), т.к. X+Y > Y, а предыдущие значения (X,Y) были (X-Y,Y)

2) Если Y>X, то последней была операция Y (Y:=X+Y), т.к. X+Y > X, а предыдущие значения (X,Y) были (X,Y-X)
3) X=Y только в самом начале, в дальнейшем X ≠ Y, т.к. X>0 и Y>0, то X+Y>X и X+Y>Y

Из этих свойств следует, что при заданных X и Y можно однозначно восстановить программу на языке XY, которая к ним привела. И, таким образом, задача сводится к перебору всех Y от 1 до N-1, вычислении строки-программы на языке XY, которая привела к (X=N,Y), и выборе среди всех этих строк самой короткой (а среди равных по длине – первой в алфавитном порядке).

Для ускорения работы программы, можно при восстановлении предыдущих значений X, Y вместо X:=X–Y (Y:=Y-X) использовать X:=X mod Y (Y:=Y mod X) – аналогично Евклидову алгоритму нахождения НОД.

Нужно учесть один частный случай – если в какой-то момент получится X=Y (или X=0 или Y=0), то значит для заданной пары (X,Y) не существует программы на языке XY, которая к ним привела бы, т.е. такую пару надо игнорировать.