

Задача 1. Замок

Имя входного файла: lock.in
Имя выходного файла: lock.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 Мбайт

Входной замок на двери столовой лагеря «Байтик» устроен следующим образом: если введено натуральное число, которое меньше 100 или больше 999, — на дисплее устройства появляется надпись «FALSE». Если у введенного числа все цифры различные или сумма цифр числа равна 13, — появляется надпись «ENTER» и можно войти. В остальных случаях появляется надпись «LOCK».

Вам необходимо составить программу, которая для заданного целого положительного числа n указывает надпись на дисплее замка.

Формат входного файла

В единственной строке записано одно число n ($1 \leq n \leq 10000$).

Формат выходного файла

Выходной файл должен содержать надпись на дисплее замка для заданного числа n .

Пример входных и выходных данных

lock.in	lock.out
1001	FALSE
175	ENTER
222	LOCK

Задача 2. Сломанный калькулятор

Имя входного файла: calc.in
Имя выходного файла: calc.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 Мбайт

В сломанном калькуляторе работают только две клавиши U и V . При нажатии клавиши U калькулятор увеличивает число в два раза, а при нажатии клавиши V написанное число увеличивается вдвое и к результату прибавляется 1. На экране написано число 0.

Вам необходимо составить программу, которая для заданного целого положительного числа n указывает последовательность операций калькулятора, с помощью которых из 0 можно получить число n , причем количество операций в этой последовательности должно быть минимальным.

Формат входного файла

В единственной строке записано одно число n ($1 \leq n \leq 2^{63} - 1$).

Формат выходного файла

Выходной файл должен содержать строку из символов U и V без пробелов, задающую последовательность операций калькулятора, с помощью которой из 0 можно получить число n . Если решений несколько, выведите любое из них.

Пример входных и выходных данных

calc.in	calc.out
12	VVUU

Пояснение к примеру

Последовательные нажатия клавиш V, V, U, U приводят к появлению на экране чисел 1, 3, 6, 12.

Задача 3. Футбольная команда

Имя входного файла: team.in
Имя выходного файла: team.out
Ограничение по времени: 2 секунда
Ограничение по памяти: 256 Мбайт

Говорят, что на исход матча влияет место, где играют футболисты. Но по последним данным британских учёных, не только место. Так же на результат влияют номера футболистов. Причем, чем ближе сумма номеров одной команды к сумме номеров другой, тем выше вероятность ничейного результата. Недоверчивые болельщики решили проверить теорию британских учёных для команды с произвольным количеством участников.

Вам необходимо разбить n футболистов на 2 команды так, чтобы разность между суммами номеров этих команд была наименьшей.

Формат входного файла

Первая строка содержит n ($2 \leq n \leq 20$) — количество футболистов. В следующей строке заданы n целых чисел — номера футболистов в диапазоне от 1 до 10^8 .

Формат выходного файла

Выходной файл должен содержать минимальную разность между суммами номеров противоположных команд.

Пример входных и выходных данных

team.in	team.out
4 1 3 2 2	0
5 7 1 3 4 2	1