

**Муниципальный этап всероссийской олимпиады школьников по программированию  
2019-2020 учебный год**

**7-8 класс**

**Задача 1.** Форт имеет форму  $N$ -угольника. В его углах находятся сторожевые башни, на которых могут располагаться солдаты. Полковнику надо расставить солдат на этих башнях так, чтобы с каждой стороны форта находилось ровно  $K$  солдат. Требуется написать программу, которая определяет, сколькими способами это можно сделать.

**Входные данные.**

С клавиатуры вводятся два натуральных числа  $N$  ( $3 \leq N \leq 10$ ) и  $K$  ( $1 \leq K \leq 1000$ ) разделённые пробелом.

**Выходные данные.**

На экран выводится искомое количество способов расстановки.

Примеры работы программы:

Ввод	Вывод
4 3	4
3 2	1
3 1	0

Пояснение к первому примеру. Полковник может на двух противоположных башнях расположить по 3, 2, 1 или 0 солдат. На двух других башнях он тогда должен расположить соответственно по 0, 1, 2 или 3 солдат. Всего – 4 способа.

**Задача 2.** Однажды одному российскому туристу довелось совершить путешествие на японском круизном лайнере. Как-то он с удивлением обнаружил, что на лайнере полностью отсутствуют каюты, номера которых содержат цифру 4. То есть, например, после каюты №13 следовала каюта №15, а после каюты №39 – сразу каюта №50. (Это странное обстоятельство объясняется тем, что цифра 4 похожа на японский иероглиф, ассоциирующийся с несчастьем.) Туристу стало интересно, сколько всего кают имеется на лайнере. Он выяснил, что номер последней каюты равен  $N$ . Требуется написать программу, которая по введённому номеру  $N$  определяет, сколько на лайнере было кают на самом деле.

**Входные данные.**

С клавиатуры вводится натуральное число  $N$  ( $1 \leq N \leq 2000000000$ ).

**Выходные данные.**

На экран выводится искомое количество кают на лайнере.

Примеры работы программы:

Входные данные	Выходные данные
13	12
50	36

**Задача 3.** В одном игровом развлекательном телешоу с участием нескольких команд игроков придумали новое суперзадание «Пересечение рва». Команда игроков находится на краю рва с водой и её задача состоит в том, чтобы обеспечить переход хотя бы одного своего игрока на другой край рва. Ров невозможно перепрыгнуть, но поперёк его расставлены в цепочку  $N$  одинаковых опорных платформ. Каждая платформа способна выдержать не более трёх игроков. Любой игрок может самостоятельно прыгнуть с края рва на первую платформу. С первой платформы можно прыгнуть на вторую платформу, со второй – на третью и т.д., с  $N$ -ой платформы можно прыгнуть на другой край рва. Однако совершать прыжки с платформ возможно лишь с помощью специального приспособления, установленного на каждой платформе.

Приспособление устроено так, что для приведения его в действие необходимо участие трёх игроков. Причём в результате его действия один игрок перепрыгивает на следующую платформу, а два других игрока падают в ров и вынуждены затем вплавь вернуться на исходный край рва. Таким образом, выполнение суперзадания становится очень непростой задачей, требующей совершения достаточно большого количества прыжков. Ваша задача – определить общее количество прыжков, необходимое для выполнения суперзадания.

Требуется написать программу, которая по заданному числу  $N$  определяет общее количество прыжков, необходимое для выполнения суперзадания. Считать, что в команде имеется необходимое количество игроков.

**Входные данные.**

С клавиатуры вводится натуральное число  $N$  ( $1 \leq N \leq 30$ ).

**Выходные данные.**

На экран выводится количество прыжков, которое необходимо совершить для выполнения суперзадания.

Пример работы программы:

Входные данные	Выходные данные
1	4
2	13

Пояснение ко второму примеру. Для выполнения суперзадания достаточно 5 игроков. Вначале все 5 игроков находятся на исходном краю рва. Последовательность дальнейших передвижений игроков отражена в следующей таблице.

Исходный край рва	1 платформа	2 платформа	Другой край рва	Количество прыжков
5	–	–	–	0
2	3	–	–	3
4	–	1	–	4
1	3	1	–	7
3	–	2	–	8
–	3	2	–	11
2	–	3	–	12
4	–	–	1	13

**Задача 4.** Как-то Пете попала старая тетрадь его дедушки, в которой оказалась таблица все натуральных чисел от 2 до 100, взятых во всех степенях от 1 до 4. Таким образом, в таблице было 4 колонки и 99 строк. Дедушка составил её, когда ещё был одарённым школьником. Все вычисления он выполнил вручную, так как в то время персональных компьютеров ещё не было. Пете захотелось проверить таблицу. Помогите ему сделать это.

Напишите программу, которая по заданному натуральному числу  $N$  проверяет правильность вычисления степеней чисел от  $N^1$  до  $N^4$ . Поскольку Петя не так трудолюбив, как его дедушка, то он решил вводить в программу все степени одной строки таблицы без разделения их пробелами или какими-нибудь другими символами. Ваша программа должна принимать на вход строку, составленную из записи степеней какого-либо числа, находить и печатать позицию первого ошибочного символа введённой строки.

**Входные данные.**

Сначала с клавиатуры вводится число  $N$  ( $2 \leq N \leq 100$ ). Затем с клавиатуры вводится строка из десятичных цифр длиной не более 30 символов.

**Выходные данные.**

Если введённая строка содержит записи чисел от  $N^1$  до  $N^4$ , то есть ошибки в строке отсутствуют, программа должна вывести на экран 0. В противном случае на экран выводится позиция первой ошибочной цифры. Если в строке не окажется ошибочных цифр, но сама строка

будет короче, чем должна быть, следует вывести позицию первой отсутствующей цифры. Если в аналогичной ситуации строка будет длиннее, чем должна быть, следует вывести позицию первой лишней цифры.

*Примеры работы программы:*

Входные данные	Выходные данные
2 24816	0
2 24576	3
2 2481	5
2 248163	6
24 2457614	7

