

Домашняя работа №3. Циклы

Код решения каждой задачи представьте в виде отдельной программы (файла `.cpp`), если не сказано иное. На проверку отправьте zip архив со всеми файлами решений. В архиве не должно быть посторонних файлов — например, `.idea`, `CMakeLists` и т.п.

Задание 1. Разминка

Код этого задания можно сдавать в произвольной форме. Например, в текстовом документе с фрагментами кода.

1.1

Реализуйте цикл, который совершает ровно `n` итераций (`n >= 0`). При этом запрещено создавать и использовать любые другие переменные, кроме `n`.

1.2

Помимо знаковых типов целых чисел (`int`) существуют также беззнаковые (`unsigned int`), которые умеют хранить только неотрицательные значения. Их использование приводит ко многим проблемам и запрещается некоторыми стайлгайдами (подробнее на [Learn C++](#)) Однако для этого задания вам нужно представить себя в мире, где есть только беззнаковые типы.

Реализуйте цикл `for` по индексу `unsigned int i`, который пробегает значения от `n-1` до `0` включительно. При этом изменять значение индекса разрешено только внутри `()` цикла `for`, а также запрещено использование `break` и `continue`.

▼ Подсказка

Чем отличается `--i` и `i--`?

Задание 2. Гипотеза Коллатца

Рассмотрим следующее преобразование:

$$f(n) = \begin{cases} n/2 & \text{if } n\%2 == 0 \\ 3n + 1 & \text{if } n\%2 == 1 \end{cases}$$

Возьмём число n и построим последовательность используя это преобразование: $a_0 = n$, $a_i = f(a_{i-1})$, $i \geq 1$. Эта последовательность может либо рано или поздно прийти в единицу, либо нет (например, заиклиться или постоянно генерировать новые числа).

Пример для $n = 11$:

$11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Ваша задача — для заданного числа n проверить, обращается ли в единицу последовательность, начинающаяся с числа n . Обратите внимание, что если ответ «нет», то программа должна выдать такой ответ, а не заиклиться. Можно считать, что достаточно проверить 100 000 первых шагов.

Задание 3. Валидация кредитной карты

Вводится номер кредитной карты (16-значное число). Проверьте, является ли он корректным. Алгоритм определения корректности можно найти, например, [здесь](#).

Задание 4. Количество точек решетки

Для заданного n найдите количество целых неотрицательных пар x и y таких, что $x^2 + y^2 < n^2$, пользуясь только операциями с целыми числами.

Вы получите дополнительные баллы, если ваше решение совершает линейное от n количество операций. То есть существует такая константа C , что количество операций не превосходит Cn для любого n .

Задание 5. Ох уж эти поляки

Рассмотрим следующую последовательность:

$$\begin{aligned}a_0 &= 1 \\ a_{i+1} &= 2a_i \oplus a_i\end{aligned}$$

Где \oplus — операция хог целых чисел.

Сгенерируйте несколько первых чисел этой последовательности и выведите их в двоичной системе счисления, по одному в строке. Для наглядности можете вывести '■' вместо '0'. Как называется полученная фигура? Ответ приведите в файле с решением.

Попробуйте сгенерировать первые 60 строк этой фигуры. Получилось? А 120? Вместе с двоичным представлением выведите десятичное обычным оператором вывода. Объясните наблюдаемый эффект.

Для получения дополнительных баллов выведите как можно больше корректных строк фигуры. Ещё больше баллов, если ваш код сможет вывести 1000 строк и не будет использовать массивы, строки или другие контейнеры.