

В этом задании нужно реализовать параллельный подсчёт количества счастливых чисел с помощью стандартной библиотеки C++. Целое положительное число k называется *счастливым*, если (варианты):

- 1) Залевский Александр: цифры в десятичной записи числа k идут по невозрастанию (3, 71, 99333100, ...);
- 2) Латышев Артём: цифры в десятичной записи числа k , стоящие на нечётных позициях, делятся на 3, а на чётных – не делятся на 3 (3, 92376, 320232, ...);
- 3) Юхимчук Александр: каждые две цифры, стоящие рядом в десятичной записи числа k , отличаются друг от друга не более чем на 2 (3, 53, 644653224, ...).

1. Напишите многопоточную программу, которая находит количество счастливых чисел от 1 до n включительно. На вход подаются два целых числа n и t . Вычисления необходимо проводить в t потоках. Выведите подсчитанное количество в `std::cout`.

2. Замерьте время работы программы при n не меньше 10^{10} (**10 в 10-й степени**), $t=1, 2, 3, \dots, k, k+1, k+2$, где k – количество ядер процессора на вашем компьютере. Сравните результаты.

3. Добавьте в программу вывод сообщений о текущем состоянии подсчёта. Каждые 5 секунд выводите, сколько чисел обработано каждым из рабочих потоков. Подумайте, как это реализовать с минимальными накладными расходами.

Важно: а) никакой рабочий поток не должен блокировать подсчёты остальных рабочих потоков; б) не должно возникать состояния гонки данных (data race) при обновлении счётчиков; в) интервал должен быть ровно 5 секунд, в него включается и время на вывод тоже.

Для реализации параллельных вычислений пользуйтесь только стандартными средствами C++11 и новее. Детали реализации, интерфейс, выбор разбиения отрезка чисел $[1, n]$ по потокам и прочее – на ваше усмотрение.

P. S. Задачу можно решать и более эффективными способами, например динамическим программированием. Но в этом задании требуется реализовать подсчёт именно проверкой каждого числа от 1 до n по отдельности.