

ОПИСАНИЕ ЗАДАНИЯ

В этой лабораторной работе от вас требуется разработать программу поиска чисел Фибоначчи с использованием динамического подхода. Динамический подход к разработке ПО (динамическое программирование) подразумевает следующее: проделав какие-либо вычисления, вы не забываете, не игнорируете, не выбрасываете их результат. Вы его сохраняете в надежде, что в следующий раз, когда снова нужно будет проводить те же вычисления, вы можете не тратить на них время, а вместо этого выдать уже готовый, посчитанный результат.

Реализация динамического подхода подразумевает, что вы сохраняете посчитанные числа в массив. Если вновь встретится необходимость вывести уже посчитанные числа, они не будут заново рассчитываться, а будут взяты из массива. Если же потребуется рассчитать новые значения, они должны быть добавлены в массив.

В процессе вывода вы должны визуально отделить числа, взятые из массива, от чисел, которые были рассчитаны.

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ БАЗОВЫЙ УРОВЕНЬ

Три плюса за лабораторную работу:

1. считать с клавиатуры целое число N , вывести на экран все числа Фибоначчи, не превышающие N ;
2. зациклить ввод: выполнять действия, описанные в п.1 пока не будет введено значение N , равное 0;
3. реализовать динамический подход.

На базовом уровне массив берите статичный, объявите массив на 1000 элементов, этого вам вполне хватит. Выделять и выделять память не нужно.

Выполнение лабораторной работы на базовом уровне оценивается в итоге в 8 баллов максимум. Лабораторная работа считается выполненной, если выполнены задания на все три плюса.

Пример

Вводится число: 4

В программе нет заранее посчитанных чисел, первый расчёт, программа считает числа и выводит ответ. Эти числа складываются в массив.

Вывод: 0 1 1 2 3

Вводится число: 2

Программа смотрит на массив и понимает, что все числа Фибоначчи до этого числа уже были посчитаны. Смысла тратить время на расчёт нет, просто выводятся необходимые значения из массива.

Вывод: *0 *1 *1 *2.

Здесь символ звёздочка показывает, что число уже было посчитано и взято из массива, смотрите на последний абзац описания задания. Как показать, что эти числа уже посчитаны – дело ваше, вы можете сделать это иначе.

Вводится число: 10

Программа смотрит на свой массив и понимает, что чисел в массиве недостаточно. Тогда она сначала выводит все имеющиеся (уже посчитанные) числа, затем досчитывает необходимые и снова складывает их в массив.

Вывод: *0 *1 *1 *2 *3 5 8

За что снимаются баллы на базовом уровне

1. код не отформатирован, нагромождение конструкций: -1 балл
2. вы не можете пояснить, как работает код с первого раза: -2 балла (но вы всё равно должны будете в итоге объяснить, как он работает, иначе лаба не будет зачтена)

ЗАДАНИЕ ДЛЯ ВЫБРАВШИХ ПОВЫШЕННЫЙ УРОВЕНЬ

Необходимо выполнить все задания на минимум. После этого необходимо выполнить дополнительные задания, приведённые ниже.

Эти задания выполняются последовательно.

- **Плюс балл.** Посчитать, сколько времени суммарно было затрачено на работу программы. Время выводить в нормальном виде: сколько минут, сколько секунд.
- **Плюс балл.** Работу с массивами реализовать через динамическую память. При необходимости – довыделять память под массив.

За что снимаются баллы на повышенном уровне

1. код не отформатирован, нагромождение конструкций: -1 балл
2. нет нормально оформленного интерфейса общения с пользователем (текст должен быть понятным, либо русский, либо английский (предпочтительно)): -1 балл.
3. вы не можете пояснить, как работает код с первого раза: -4 балла (но вы всё равно должны будете в итоге объяснить, как он работает, иначе лаба не будет зачтена)