

1. Напишите на языке ассемблера подпрограмму, вычисляющую n -й элемент последовательности Фибоначчи:

- а. Заголовок функции (подпрограммы) находится в файле `main.c` в папке Z1.
- б. Подпрограмма возвращает код ошибки в качестве возвращаемого значения — ошибка возникает, если при добавлении происходит переполнение (для практики можно расширить задачу, чтобы ошибка возникала и при передаче невалидного значения). n ; подпрограмма возвращает 0, если ошибки не произошло, и все остальное, если она произошла
- в. Подпрограмма возвращает найденный n -й элемент массива через параметр *беззнаковое целое* вырезать* передано по адресу

2. Напишите подпрограмму на языке ассемблера, которая вычисляет n -й элемент последовательности Фибоначчи, используя **рекурсивный алгоритм**. (Более подробную информацию о рекурсивном алгоритме Фибоначчи можно найти по ссылке <https://medium.com/launch-school/recursive-fibonacci-method-explained-d82215c5498e>, но основная идея состоит в том, что n -й элемент последовательности вычисляется по формуле:

$F(n) = F(n-1) + F(n-2)$, для $n > 2$, если последовательность индексируется, начиная с 1, и $F(1) = F(2) = 1$:

- а. Заголовок функции (подпрограммы) находится в файле `main.c` в папке Z2.
- б. Подпрограмма возвращает результаты в обратном порядке, чем задача 1: найденный n -й элемент возвращается как возвращаемое значение подпрограммы, а ошибка возвращается через параметр, передаваемый по адресу.

в. Помощь—В этой подпрограмме есть три важных блока:

- i. Установка условия выхода из рекурсии — его следует ставить вверху, чтобы можно было вовремя остановить рекурсию; условие отражается на проверке того, присутствует ли в данный момент $n > 2$ и следует ли вводить дальнейшую рекурсию
- ii. Вызов подпрограммы для $F(n-1)$
- iii. Вызов подпрограммы для $F(n-2)$ и суммирование результата с результатом вызова $F(n-1)$

д. Примечание: решение, доступное в ACS, не содержит кода, учитывающего ошибки, поэтому его необходимо расширить, чтобы оно соответствовало прикрепленному файлу `.c`.

3. Напишите подпрограмму на ассемблере, которая покомпонентно складывает две строки `аиб` и результат сложения помещается в массив c ($c[0] = a[0] + b[0]$, $c[1] = a[1] + b[1]$, ..., $c[n] = a[n] + b[n]$):

- а. Элементы всех трех массивов имеют двойную точность.
- б. Заголовок функции (подпрограммы) находится в файле `main.c` в папке Z3.
- в. Все три строки передаются подпрограмме как параметры, передаваемые по адресу.
- д. Подпрограмма возвращает код ошибки в качестве возвращаемого значения; возникает ошибка, если при добавлении происходит перерасход; процедура возвращает 0, если ошибки не было, и что-нибудь еще, если она была.

Примечание: вы можете запустить тестовый сценарий, просто передав ему свою подпрограмму, то есть файл `.S` (нет необходимости передавать файл `.c`) — `./test.sh my_solution.S`. Если вы получили сообщение об ошибке, сообщающее, что у вас нет прав на выполнение скрипта, откройте терминал в папке, где находится тестовый скрипт, и введите команду:

`chmod +x test.sh`