1

# 0.0.1 Задание 1

Используем готовую реализацию Forthress чтобы немного привыкнуть к новому языку программирования.

1. Изучите файл README.md в репозитории Forthress.
2. Изучите программы из предыдущей секции.
3. Напишите программу, проверяющую число на четность.

Вы можете подавать программу на вход скрипту start и она будет выполняться при запуске. Слив его содержимое с потоком ввода с помощью cat вы можете получить эффект аналогичный предзагрузке этого файла.

1. Напишите программу, проверяющую число на простоту.
2. Напишите программу так, чтобы она выделяла с помощью allot ячейку в памяти, записывала туда результат и возвращала её адрес.

Обратите внимание, вы не обязаны делать все эти действия в одном и том же слове. Делайте максимально короткие слова и комбинируйте их.

1. Создать строку в куче в Forthress можно с помощью следующего синтаксиса: m" string". Слово prints печатает строку по указателю. Память в куче должна освобождаться с помощью heap−free.

С помощью heap−show можно получить диагностическую информацию о куче.

С помощью ? можно получить диагностическую информацию о любом числе или адресе. Например, попробуйте ввести:

42 ? ' dup ? m" hello, world" ?

1. Прочитайте про слова c@, c!. Напишите слово, которое принимает указатели на две строки и возвращает их конкатенацию. Память под строку-результат вы можете выделить с помощью слова heap−alloc, которое принимает количество байт для выделения и Протестируйте полученное слово. Нет ли утечки памяти?

Вторая часть задания организована по вариантам. Ваш вариант подсчитывается с помощью следующей функции, применённой к вашей фамилии:

( str − num )

: string−hash

0 >r ( init accumulator ) repeat

dup c@ ( stacks: str char, acc ) dup if ( not end of the line ) r> 13 \* + 65537 % ( iteration of hash computations )

>r 1 + 0

else ( end of line ) drop drop r> 1

then

until

;

2

Это слово уже определено в файле hash.frt. Номер варианта подсчитывается так: от вашей фамилии берётся хэш, затем вы берёте значение хэша по модулю 3. Варианты:

1. Напишите слово, которое для положительного числа построит последовательность Коллатца.

Последовательность Коллатца строится так:

* + Если текущий член последовательности делится на два, то делим его пополам.
  + Иначе умножаем его на 3 и прибавляем 1.

Последовательность Коллатца всегда приходит к единице, но никто не знает, почему.

1. Напишите слово, которое проведёт проверку числа на примарность.

Примарным называется число, которое представляется в виде произведения простых без повторов. Например, число 4 не примарное, т.к. 4=22; 20 – не примарное число, т.к. 20=22 · 51; 15 – примарное число, т.к. 15=31 · 51.

1. Напишите слово, которое найдёт произведение всех простых делителей числа (его радикал).

Например, для числа 20 это 10, т.к. 20=22 · 51, его простые делители 2 и 5.