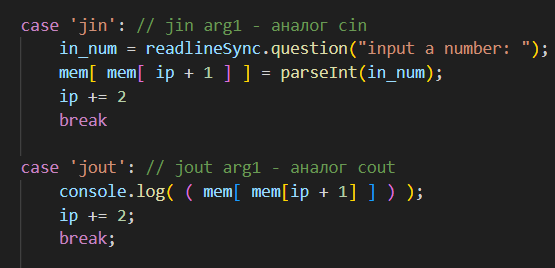
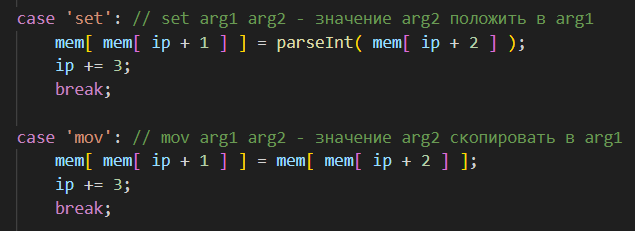
Virtual machine – написание своего низкоуровневого языка программирования. Код работает с «ячейками памяти», в которые мы заносим данные или команды. Пусть наш язык называется jpp и использует расширение «.jpp».

Перед началом работы нужно скачать модуль readline-sync:

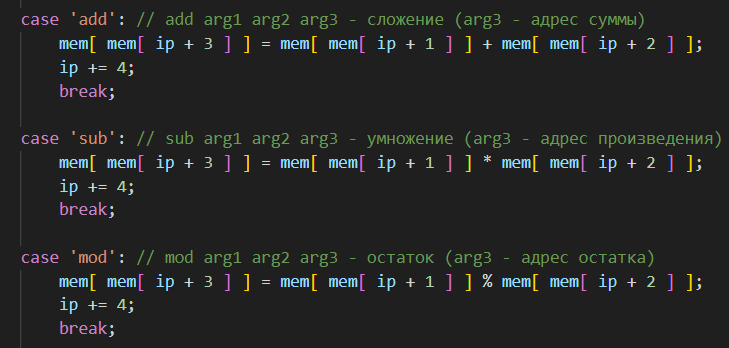
Разберём, какие команды есть:



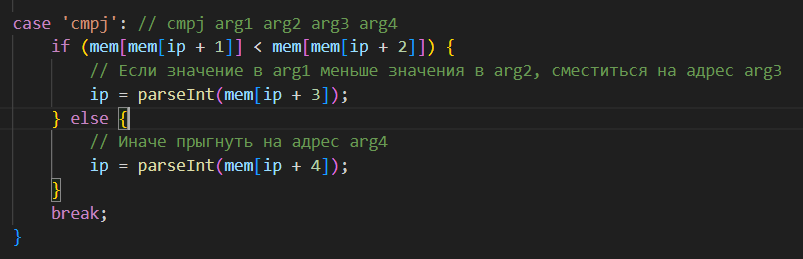
jin и jout – ввод и вывод в терминал. jin принимает число и записывает его в ячейку arg1, jout выводит лежащий в arg1 аргумент.



set – в arg1 заносится число, которое мы пишем в arg2 (например set 105 74 – в ячейке 105 лежит значение 74). mov – копирует значение arg2 в arg1 (например аргумент ячейки 11 равен 84, после mov 100 11 в ячейку 100 будет скопировано число 84).



add и sub работают по похожему принципу, значение аргумента первой ячейки складывается (умножается) со значением аргумента второй, после сумма (произведение) записывается в третью. mod ищет остаток от деления аргумента первой ячейки на аргумент второй и записывает его в третью.



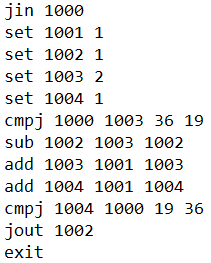
cmpj – compare jump – сравнивает значение аргумента первой ячейки со значением аргумента во второй. После, если первый меньше, перемещает нас по адресу третьей ячейки, иначе по адресу четвёртой. В особенности нужен для циклов.

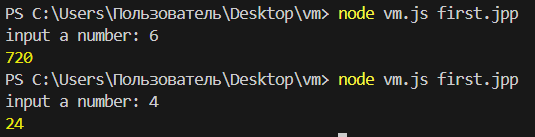
Алгоритм завершает выполнение, когда натыкается на команду exit.

Для запуска выполнения надо ввести название файла, в котором содержится алгоритм.

Разберем на примерах работу виртуальной машины.

В первом примере мы вводим число и выводим факториал этого числа.





Во втором примере находим НОД двух вводимых чисел.

