* Язык программирования. Синтаксис
  + lisp -- синтаксис языка Lisp. S-exp. Любое выражение -- expression.
* Архитектура
  + cisc -- система команд должна содержать сложные инструкции переменной длины и разрешающие адресацию как регистров, так и памяти.
* Архитектура организации памяти
  + Гарвардская архитектура.
* Control Unit
  + instr -- процессор необходимо моделировать с точностью до каждой инструкции (наблюдается состояние после каждой инструкции).
* Точность модели
  + instr -- процессор необходимо моделировать с точностью до каждой инструкции (наблюдается состояние после каждой инструкции).
* Представление машинного кода.
  + struct -- в виде высокоуровневой структуры данных. Считается, что одна инструкция укладывается в одно машинное слово, за исключением CISC архитектур.
* Ввод-вывод Ввод-вывод осуществляется как поток токенов. Есть в примере. Логика работы:
  + при старте модели у вас есть буфер, в котором представлены все данные ввода (['h', 'e', 'l', 'l', 'o']);
  + при обращении к вводу (выполнение инструкции), модель процессора получает "токен" (символ) информации;
  + если данные в буфере кончились -- останавливайте моделирование;
  + вывод данных реализуется аналогично, по выполнению команд в буфер вывода добавляется ещё один символ;
  + по окончанию моделирования показать все выведенные данные;
  + логика работы с буфером реализуется в рамках модели на Python.
* Ввод-вывод ISA - Поддержка ввода-вывода с точки зрения системы команд.
  + port -- port-mapped
* Алгоритм - prob1.
  + Multiples of 3 or 5 (Если мы перечислим все натуральные числа ниже 10, кратные 3 или 5, мы получим 3, 5, 6 и 9. Сумма этих кратных равна 23. Найдите сумму всех чисел, кратных 3 или 5, меньших 1000.)