**Лабораторной Работа 9**

### Программирование цикла. Обработка аргументов командной строки.

## Герра Гарсия Максимиано Антонио

**Содержание**

1. [Цель работы](#_bookmark0) 5
2. [Задание](#_bookmark1) 6
3. [Теоретическое введение](#_bookmark2) 7
4. [Выполнение лабораторной работы](#_bookmark3) 8
5. [Выводы](#_bookmark18) 19

[Список литературы](#_bookmark19) 20

**Список иллюстраций**

[4.1 Файл lab9-1.asm](#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

[4.2 Работа программы lab9-1.asm](#_bookmark5) . . . . . . . . . . . . . . . . . . . . 9

* 1. [Файл lab9-1.asm](#_bookmark6) 10
  2. [Работа программы lab9-1.asm](#_bookmark7) 11
  3. [Файл lab9-1.asm](#_bookmark8) 12
  4. [Работа программы lab9-1.asm](#_bookmark9) 12
  5. [Файл lab9-2.asm](#_bookmark10) 13
  6. [Работа программы lab9-2.asm](#_bookmark11) 14
  7. [Файл lab9-3.asm](#_bookmark12) 15
  8. [Работа программы lab9-3.asm](#_bookmark13) 15
  9. [Файл lab9-3.asm](#_bookmark14) 16
  10. [Работа программы lab9-3.asm](#_bookmark15) 17
  11. [Файл lab9-4.asm](#_bookmark16) 18
  12. [Работа программы lab9-4.asm](#_bookmark17) 18

# Цель работы

Приобретение навыков написания программ с использованием циклов и обра- боткой аргументов командной строки.

# Задание

* 1. Изучите примеры программ
  2. Напишите программу, которая находит сумму значений функции f(x) для x = x1, x2 , …, xn, т.е. программа должна выводить значение f(x1) + f(x2)+…+f(xn). Значения x передаются как аргументы. Вид функции f(x) выбрать из табли- цы 9.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах x.
  3. Загрузите файлы на GitHub.

# Теоретическое введение

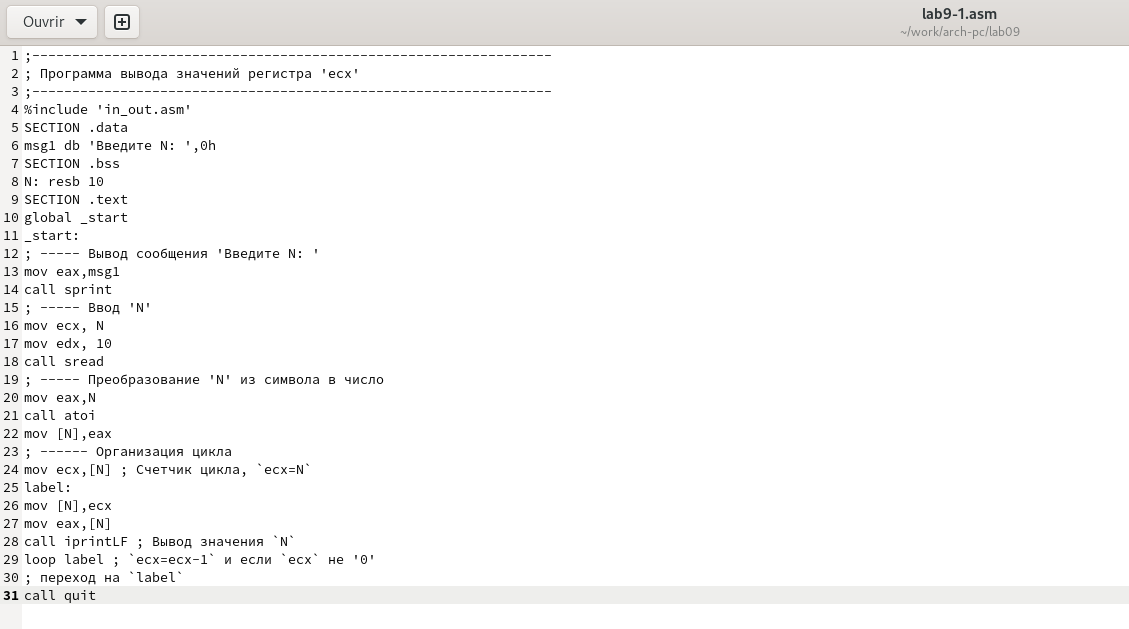
Стек — это структура данных, организованная по принципу LIFO («Last In

— First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

Для организации циклов существуют специальные инструкции. Для всех ин- струкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Иструкция loop выполняется в два этапа. Сна- чала из регистра ecx вычитается единица и его значение сравнивается с нулём. Если регистр не равен нулю, то выполняется переход к указанной метке. Иначе переход не выполняется и управление передаётся команде, которая следует сразу после команды loop.

# Выполнение лабораторной работы

* 1. Создайте каталог для программам лабораторной работы № 9, перейдите в него и создайте файл lab9-1.asm
  2. Введите в файл lab9-1.asm текст программы из листинга 9.1. Создайте ис- полняемый файл и проверьте его работу. (рис. [4.1,](#_bookmark4) [4.2)](#_bookmark5)

Рис. 4.1: Файл lab9-1.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

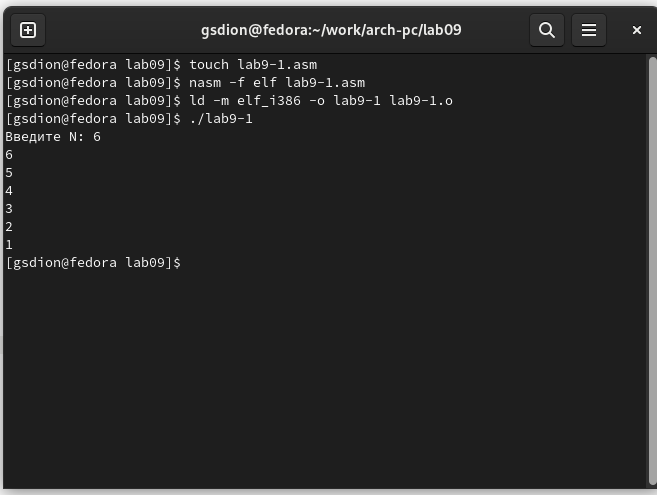
Description automatically generated

Рис. 4.2: Работа программы lab9-1.asm

* 1. Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы. Измените текст программы добавив изменение значение регистра ecx в цикле: Создайте исполняемый файл и проверьте его работу. Какие значения принимает регистр ecx в цикле? Соответствует ли число проходов цикла значению N, введенному с клавиатуры? (рис. [4.3,](#_bookmark6) [4.4)](#_bookmark7)

Программа запускает бесконечный цикл при нечетном N и выводит только нечетные числа при четном N.

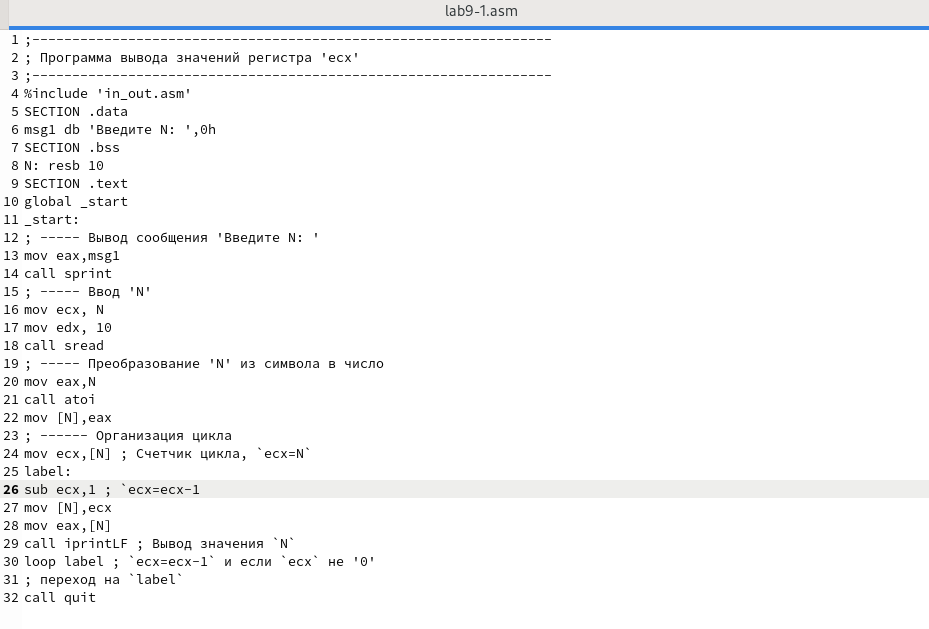


Рис. 4.3: Файл lab9-1.asm

Logo

Description automatically generated

Рис. 4.4: Работа программы lab9-1.asm

* 1. Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Внесите изменения в текст програм- мы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop. Создайте исполняемый файл и проверьте его работу. Соответствует ли в данном случае число проходов цикла значению N введенному с клавиатуры? (рис. [4.5,](#_bookmark8) [4.6)](#_bookmark9)

Программа выводит числа от N-1 до 0, число проходов цикла соответсвует N.

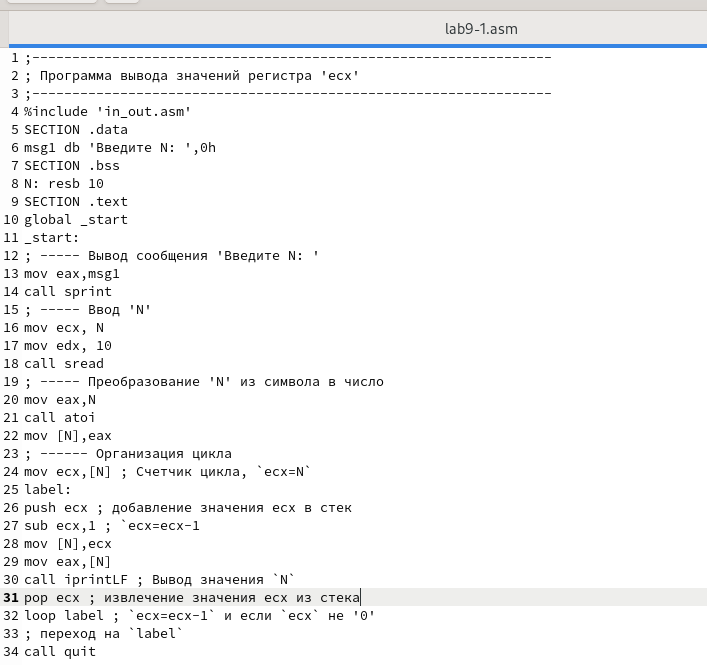


Рис. 4.5: Файл lab9-1.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

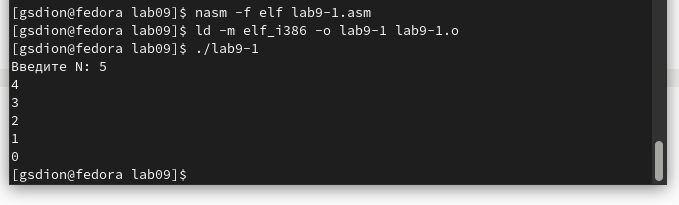
Description automatically generated

Рис. 4.6: Работа программы lab9-1.asm

* 1. Создайте файл lab9-2.asm в каталоге ~/work/arch-pc/lab09 и введите в него текст программы из листинга 9.2. Создайте исполняемый файл и запустите его, указав аргументы. (рис. [4.7,](#_bookmark10) [4.8)](#_bookmark11) Сколько аргументов было обработано программой?

Программа обработала 5 аргументов.

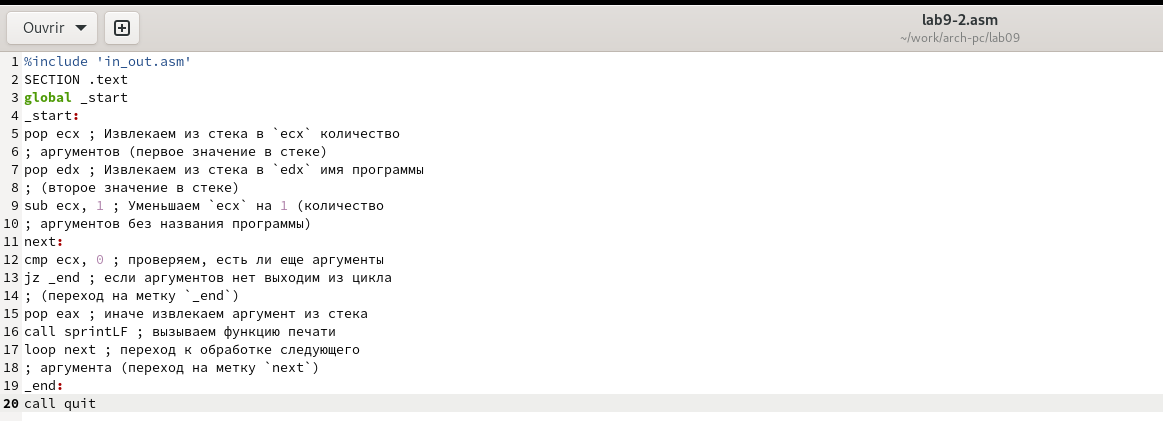


Рис. 4.7: Файл lab9-2.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

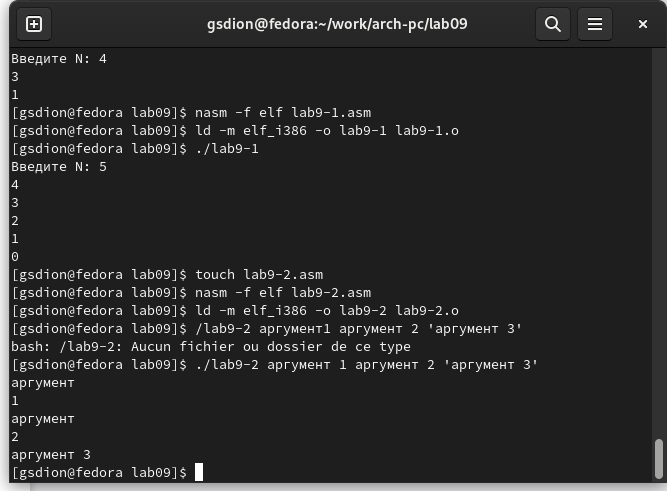
Description automatically generated

Рис. 4.8: Работа программы lab9-2.asm

* 1. Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. (рис. [4.9,](#_bookmark12) [4.10)](#_bookmark13)

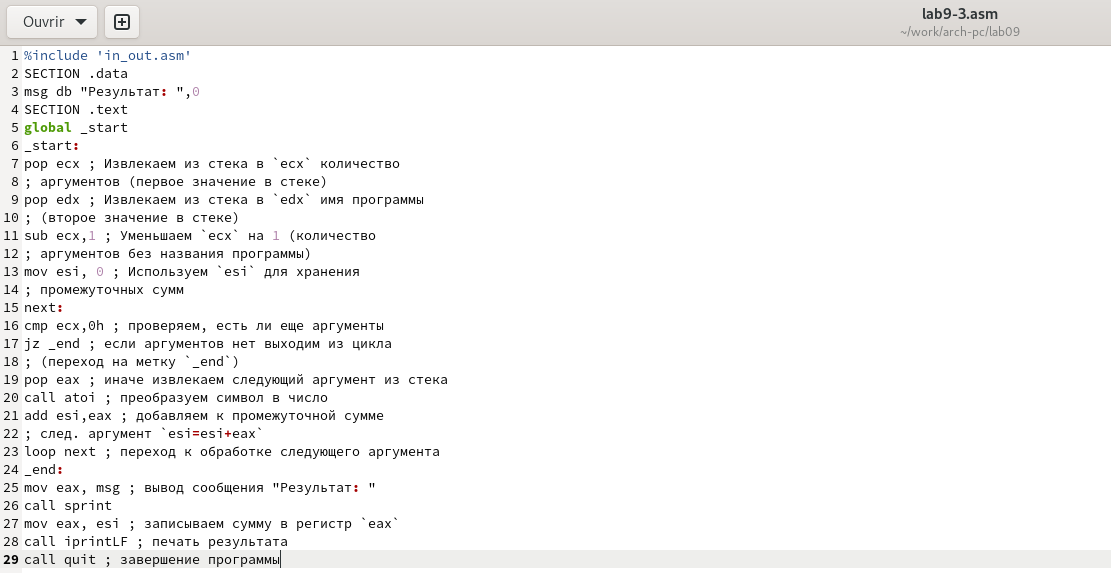


Рис. 4.9: Файл lab9-3.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

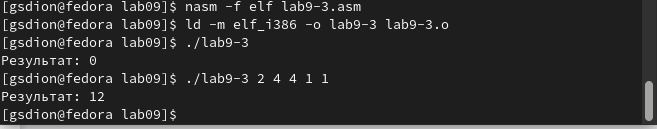
Description automatically generated

Рис. 4.10: Работа программы lab9-3.asm

* 1. Измените текст программы из листинга 9.3 для вычисления произведения аргументов командной строки. (рис. [4.11,](#_bookmark14) [4.12)](#_bookmark15)

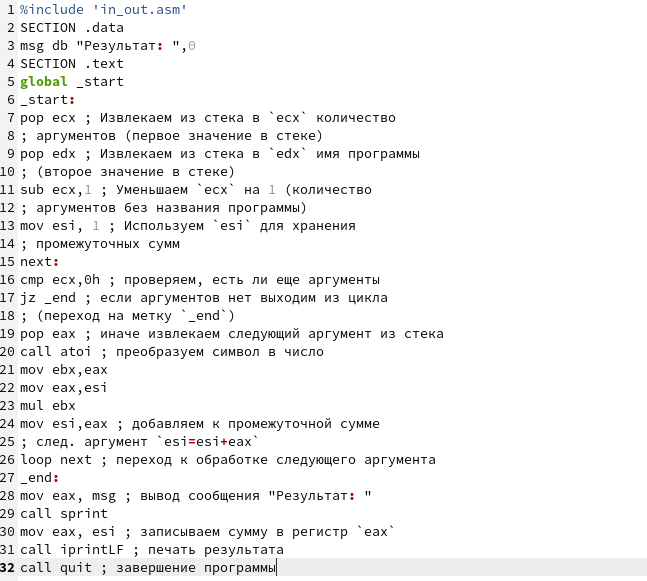


Рис. 4.11: Файл lab9-3.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

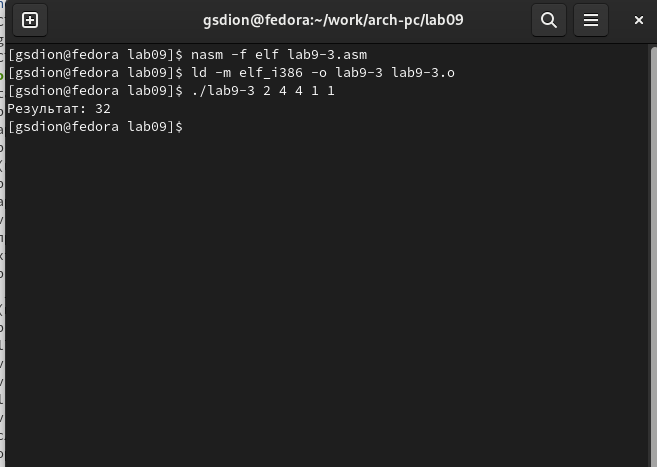
Description automatically generated

Рис. 4.12: Работа программы lab9-3.asm

* 1. Напишите программу, которая находит сумму значений функции f(x) для x = x1, x2 , …, xn, т.е. программа должна выводить значение f(x1) + f(x2)+…+f(xn). Значения x передаются как аргументы. Вид функции f(x) выбрать из табли- цы 9.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах x. (рис. [4.13,](#_bookmark16) [4.14)](#_bookmark17)

для варивнта 10 f(x) = 5(2+x)

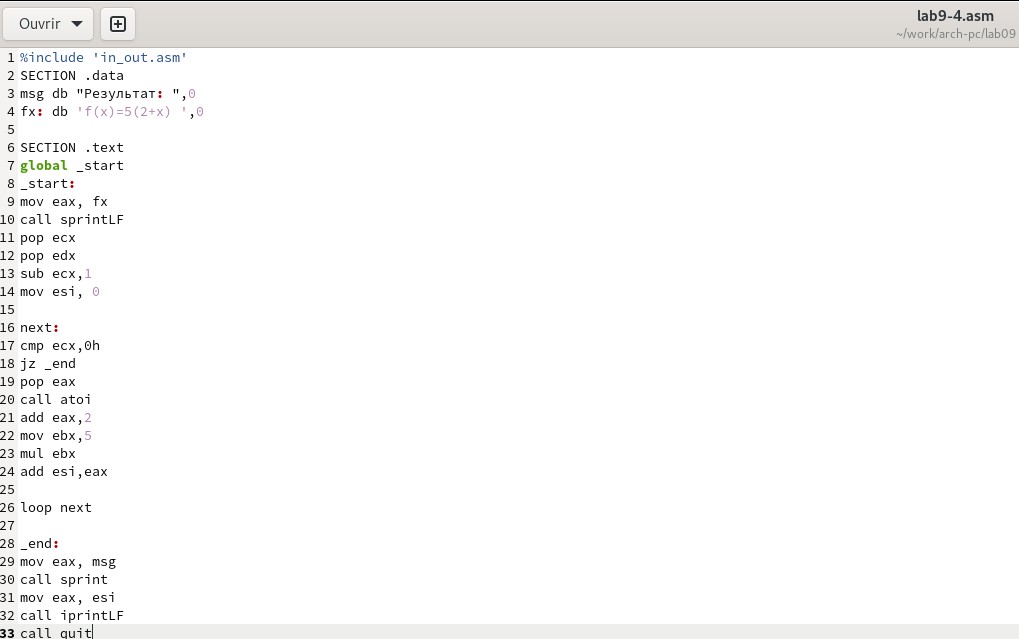


Рис. 4.13: Файл lab9-4.asm

Logo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

Description automatically generatedLogo

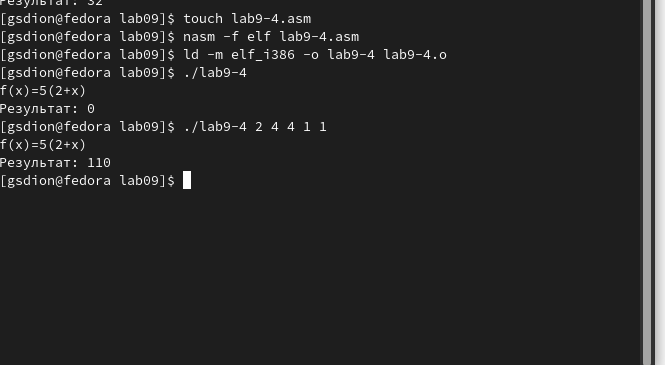
Description automatically generated

Рис. 4.14: Работа программы lab9-4.asm

# Выводы

В заключение, это позволило нам освоить работу со стеком, циклом и аргумен- тами в ассемблере Nasm.

# Список литературы

* 1. [Расширенный ассемблер: NASM](https://www.opennet.ru/docs/RUS/nasm/)