

# **Отчёта по лабораторной работе 9**

**Программирование цикла. Обработка аргументов командной строки.**

Онвудиве Виктор Чибуйке

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Выводы</b>	<b>19</b>

## Список иллюстраций

2.1	Файл lab9-1.asm . . . . .	7
2.2	Работа программы lab9-1.asm . . . . .	8
2.3	Файл lab9-1.asm . . . . .	9
2.4	Работа программы lab9-1.asm . . . . .	9
2.5	Файл lab9-1.asm . . . . .	10
2.6	Работа программы lab9-1.asm . . . . .	11
2.7	Файл lab9-2.asm . . . . .	11
2.8	Работа программы lab9-2.asm . . . . .	12
2.9	Файл lab9-3.asm . . . . .	13
2.10	Работа программы lab9-3.asm . . . . .	14
2.11	Файл lab9-3.asm . . . . .	15
2.12	Работа программы lab9-3.asm . . . . .	16
2.13	Файл lab9-4.asm . . . . .	17
2.14	Работа программы lab9-4.asm . . . . .	18

## Список таблиц

# 1 Цель работы

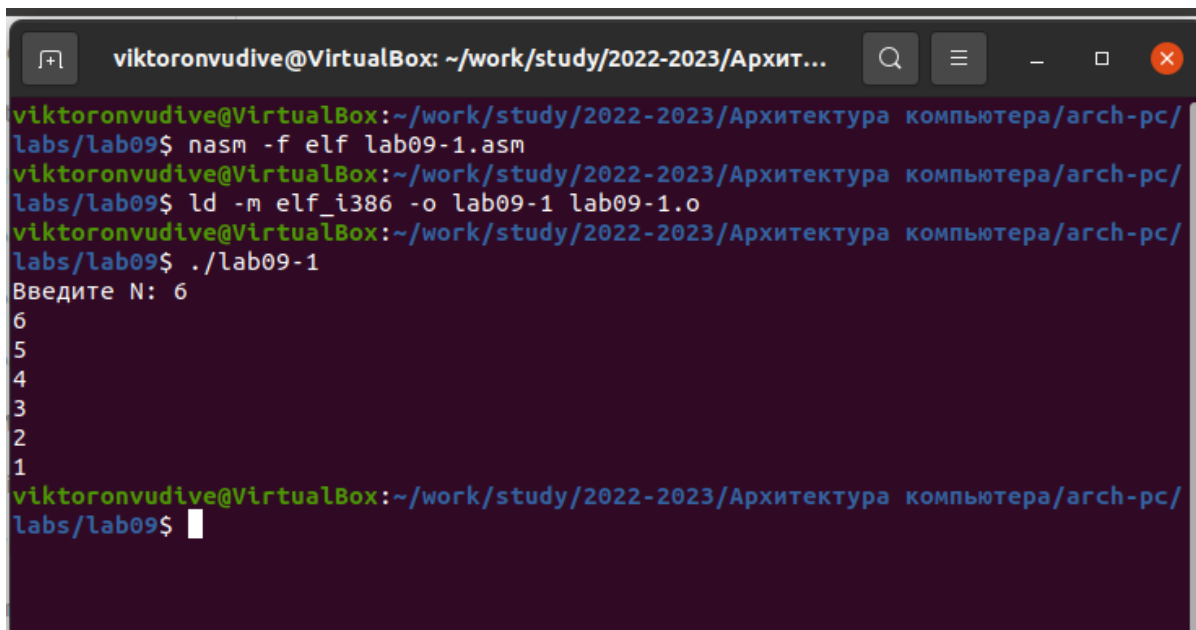
Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки..

## 2 Выполнение лабораторной работы

1. Создайте каталог для программ 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 исполняемый файл и проверьте его работу. (рис. [2.1], [2.2])

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 mov [N],ecx
24 mov eax,[N]
25 call iprintLF ; Вывод значения `N`
26 loop label ; `ecx=ecx-1` и если `ecx` не `0`
27 ; переход на `label`
28 call quit
```

Рис. 2.1: Файл lab9-1.asm



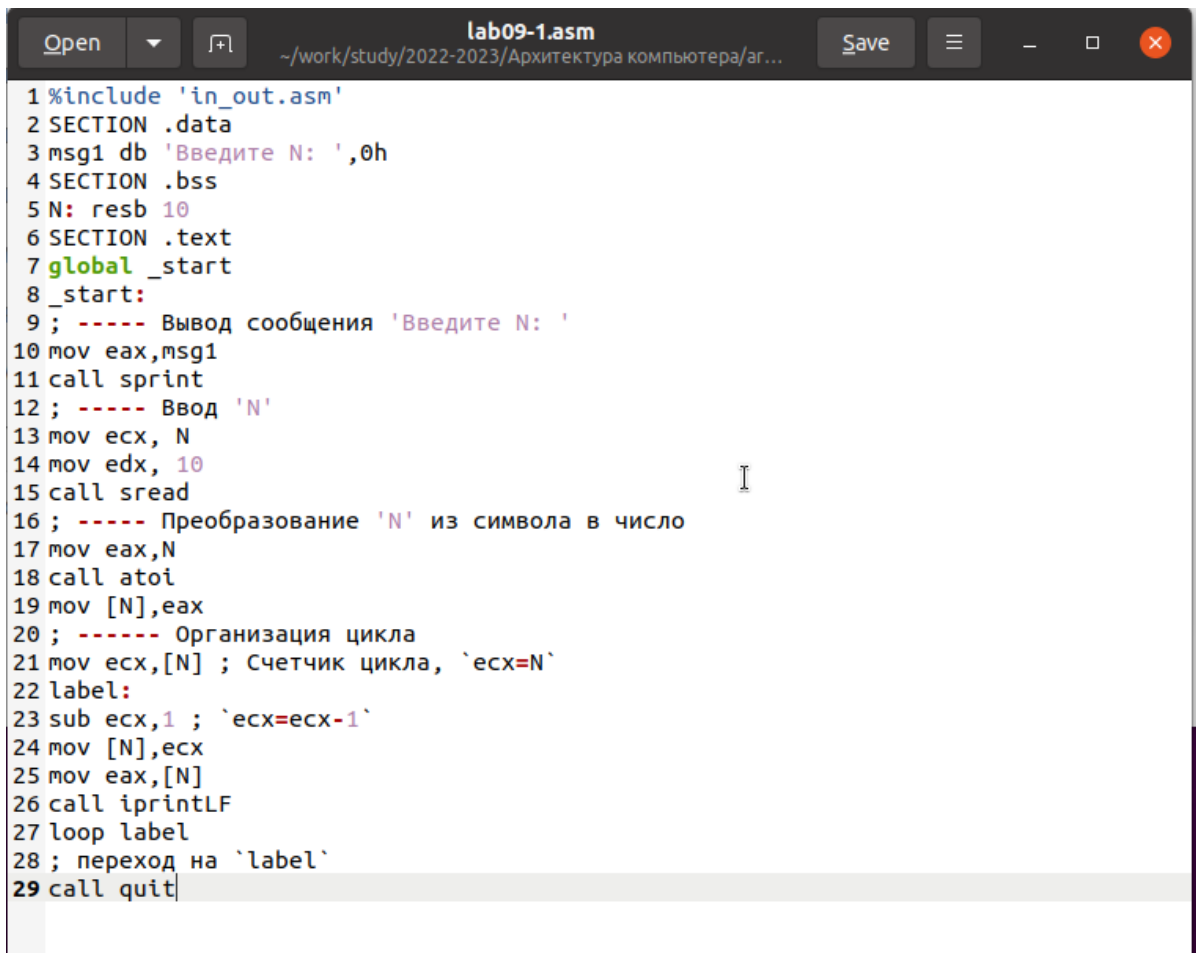
```
viktoronvudive@VirtualBox: ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab09$ nasm -f elf lab09-1.asm
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab09$ ld -m elf_i386 -o lab09-1 lab09-1.o
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab09$ ./lab09-1
Введите N: 6
6
5
4
3
2
1
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab09$
```

Рис. 2.2: Работа программы lab9-1.asm

3. Данный пример показывает, что использование регистра `ecx` в теле цикла `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. [2.3], [2.4])

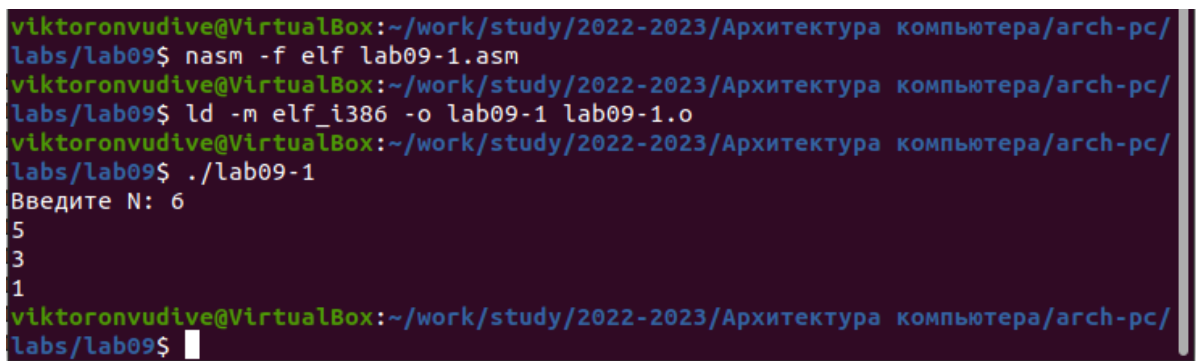
Программа запускает бесконечный цикл при нечетном `N` и выводит только нечетные числа при четном `N`.





```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 sub ecx,1 ; `ecx=ecx-1`
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF
27 loop label
28 ; переход на `label`
29 call quit
```

Рис. 2.3: Файл lab9-1.asm



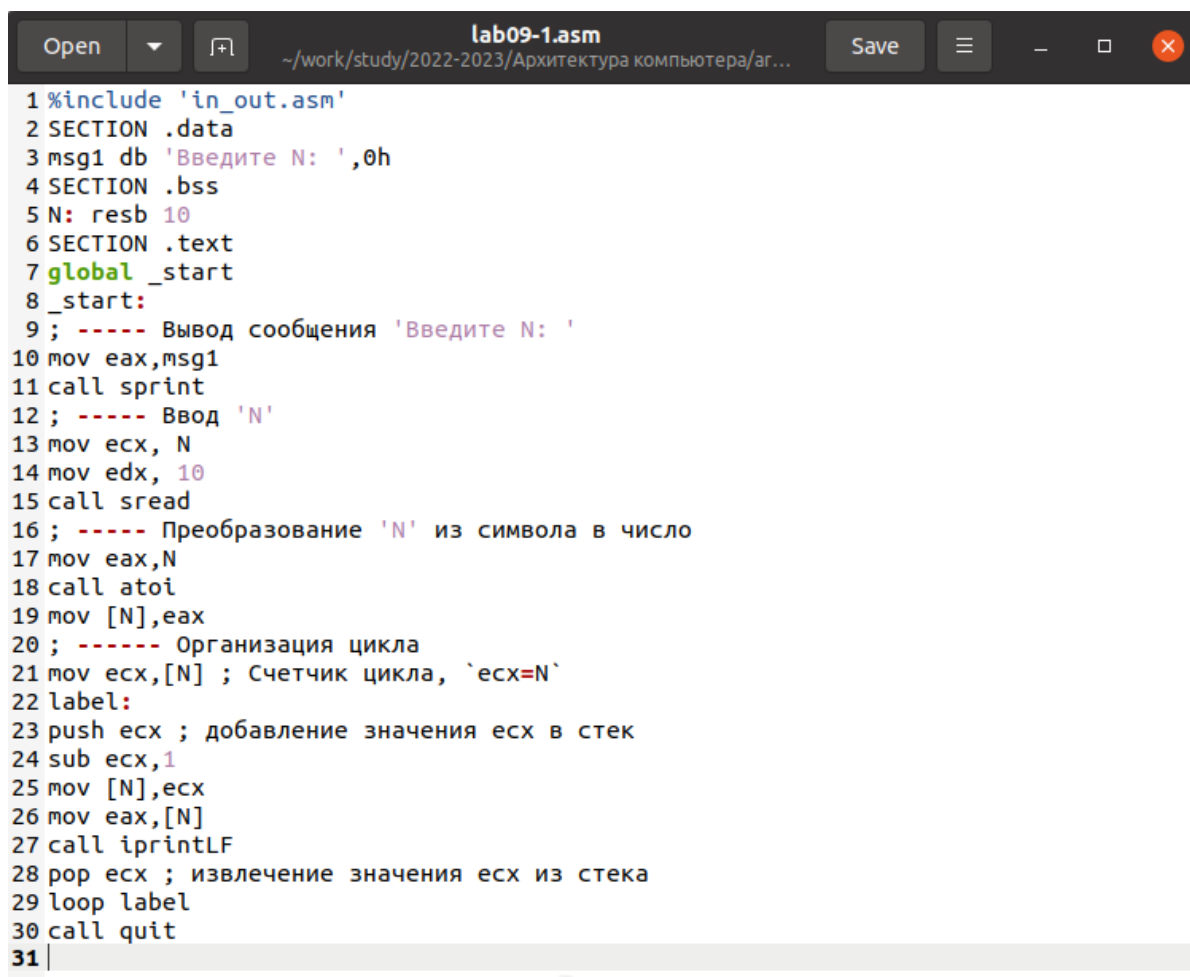
```
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ nasm -f elf lab09-1.asm
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ ld -m elf_i386 -o lab09-1 lab09-1.o
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ ./lab09-1
Введите N: 6
5
3
1
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$
```

Рис. 2.4: Работа программы lab9-1.asm

4. Для использования регистра ecx в цикле и сохранения корректности работы

программы можно использовать стек. Внесите изменения в текст программы 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. [2.5], [2.6])

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



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 push ecx ; добавление значения ecx в стек
24 sub ecx,1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF
28 pop ecx ; извлечение значения ecx из стека
29 loop label
30 call quit
31
```

Рис. 2.5: Файл lab9-1.asm

```

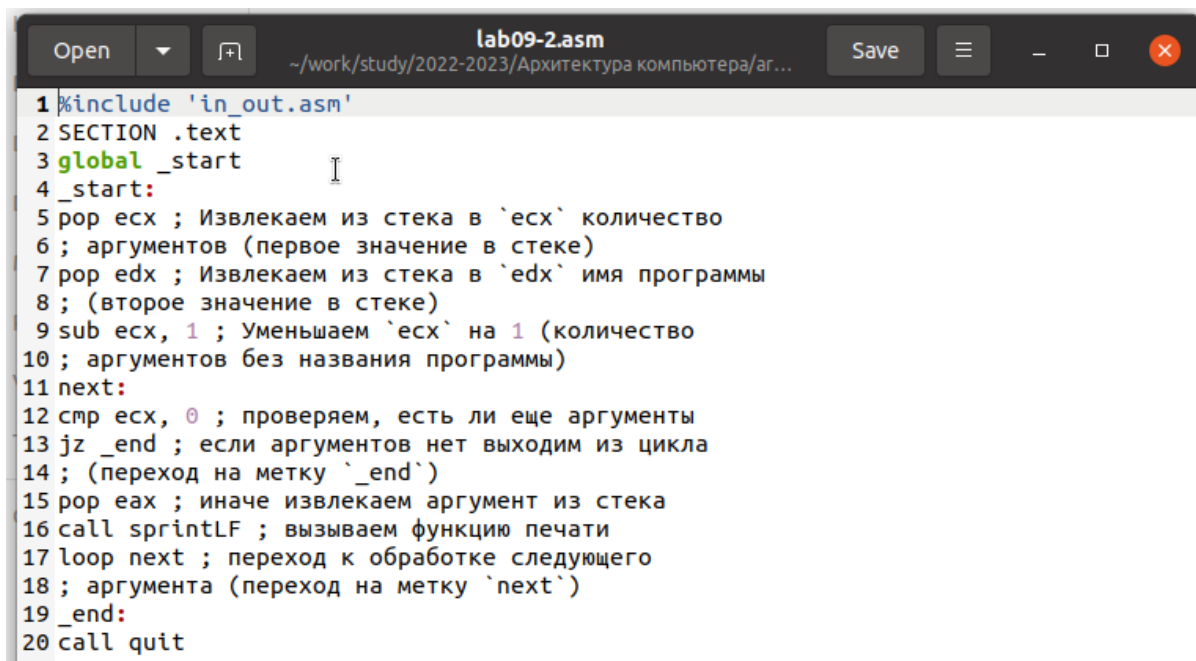
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ nasm -f elf lab09-1.asm
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ ld -m elf_i386 -o lab09-1 lab09-1.o
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$ ./lab09-1
Введите N: 6
5
4
3
2
1
0
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/
labs/lab09$

```

Рис. 2.6: Работа программы lab9-1.asm

- 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. [2.7], [2.8]) Сколько аргументов было обработано программой?

Программа обработала 5 аргументов.



```

lab09-2.asm
~/work/study/2022-2023/Архитектура компьютера/ар...
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в `ecx` количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в `edx` имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку `_end`)
15 pop eax ; иначе извлекаем аргумент из стека
16 call sprintf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit

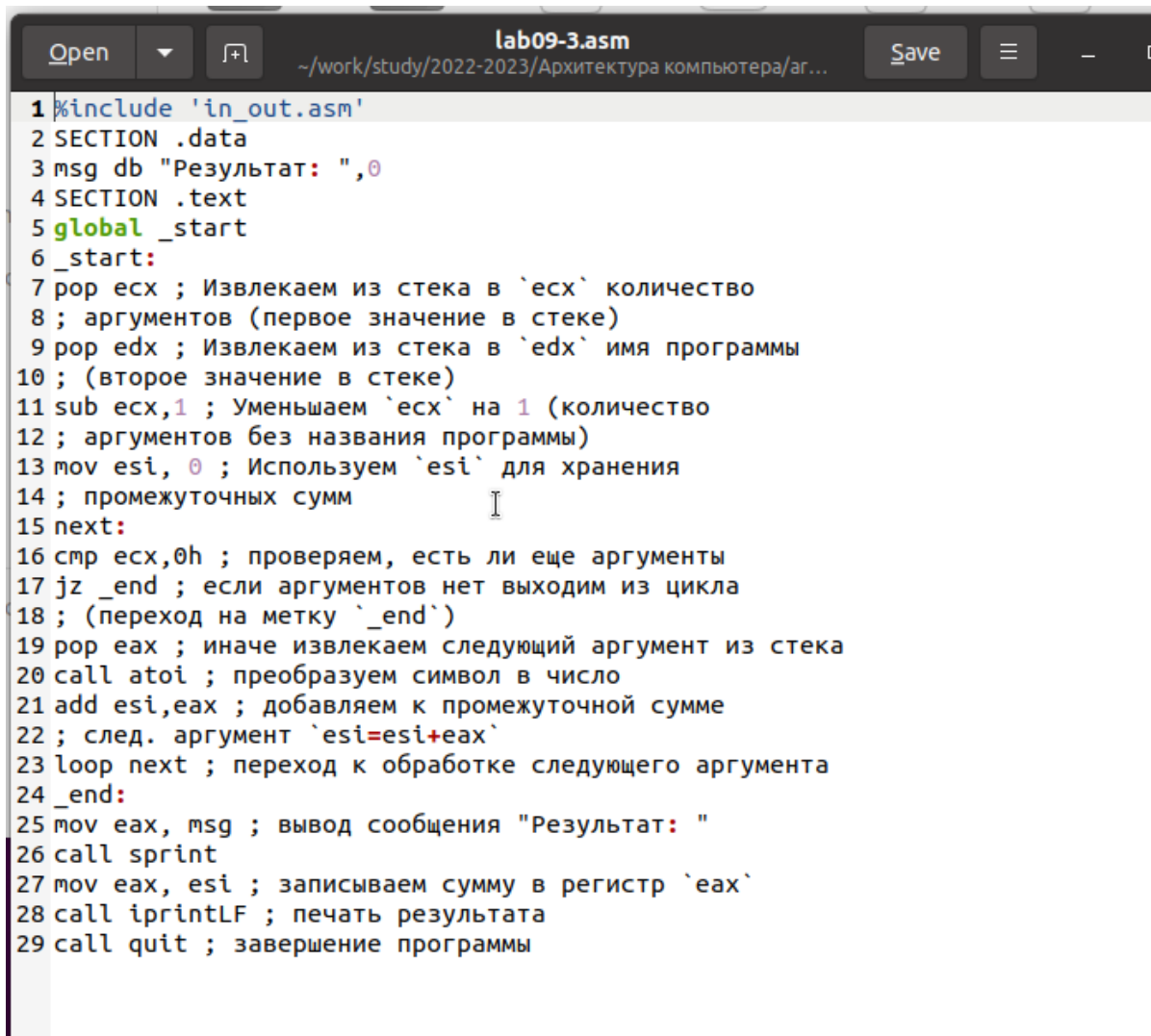
```

Рис. 2.7: Файл lab9-2.asm

```
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/  
labs/lab09$  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/  
labs/lab09$ nasm -f elf lab09-2.asm  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/  
labs/lab09$ ld -m elf_i386 -o lab09-2 lab09-2.o  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/  
labs/lab09$ ./lab09-2 argument 1 argument 2 "argument 3"  
argument  
1  
argument  
2  
argument 3  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьютера/arch-pc/  
labs/lab09$
```

Рис. 2.8: Работа программы lab9-2.asm

6. Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. (рис. [2.9], [2.10])



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7 pop ecx ; Извлекаем из стека в `ecx` количество
8 ; аргументов (первое значение в стеке)
9 pop edx ; Извлекаем из стека в `edx` имя программы
10 ; (второе значение в стеке)
11 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
12 ; аргументов без названия программы)
13 mov esi, 0 ; Используем `esi` для хранения
14 ; промежуточных сумм
15 next:
16 cmp ecx,0h ; проверяем, есть ли еще аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку `_end`)
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 add esi,eax ; добавляем к промежуточной сумме
22 ; след. аргумент `esi=esi+eax`
23 loop next ; переход к обработке следующего аргумента
24 _end:
25 mov eax, msg ; вывод сообщения "Результат: "
26 call sprint
27 mov eax, esi ; записываем сумму в регистр `eax`
28 call iprintfLF ; печать результата
29 call quit ; завершение программы
```

Рис. 2.9: Файл lab9-3.asm

```
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьюте
labs/lab09$ nasm -f elf lab09-3.asm
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьюте
labs/lab09$ ld -m elf_i386 -o lab09-3 lab09-3.o
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьюте
labs/lab09$ ./lab09-3 1 2 3
Результат: 6
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьюте
labs/lab09$ ./lab09-3 1 2 3 98 7 1 2 3
Результат: 117
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура компьюте
labs/lab09$
```

Рис. 2.10: Работа программы lab9-3.asm

7. Измените текст программы из листинга 9.3 для вычисления произведения аргументов командной строки. (рис. [2.11], [2.12])

```
1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
6 _start:
7 pop ecx ; Извлекаем из стека в `ecx` количество
8 ; аргументов (первое значение в стеке)
9 pop edx ; Извлекаем из стека в `edx` имя программы
10 ; (второе значение в стеке)
11 sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
12 ; аргументов без названия программы)
13 mov esi,1 ; Используем `esi` для хранения
14 ; промежуточных сумм
15 next:
16 cmp ecx,0h ; проверяем, есть ли ещё аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку `_end`)
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 mov ebx,eax
22 mov eax,esi
23 mul ebx
24 mov esi,eax ; добавляем к промежуточной сумме
25 ; след. аргумент `esi=esi+eax`
26 loop next ; переход к обработке следующего аргумента
27 _end:
28 mov eax,msg ; вывод сообщения "Результат: "
29 call sprint
30 mov eax,esi ; записываем сумму в регистр `eax`
31 call iprintLF ; печать результата
32 call quit ; завершение программы
33 |
```

Рис. 2.11: Файл lab9-3.asm

```

viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура ком
labs/lab09$ nasm -f elf lab09-3.asm
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура ком
labs/lab09$ ld -m elf_i386 -o lab09-3 lab09-3.o
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура ком
labs/lab09$ ./lab09-3 1 2 3
Результат: 6
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура ком
labs/lab09$ ./lab09-3 1 2 3 98 7 1 2 3
Результат: 24696
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура ком
labs/lab09$

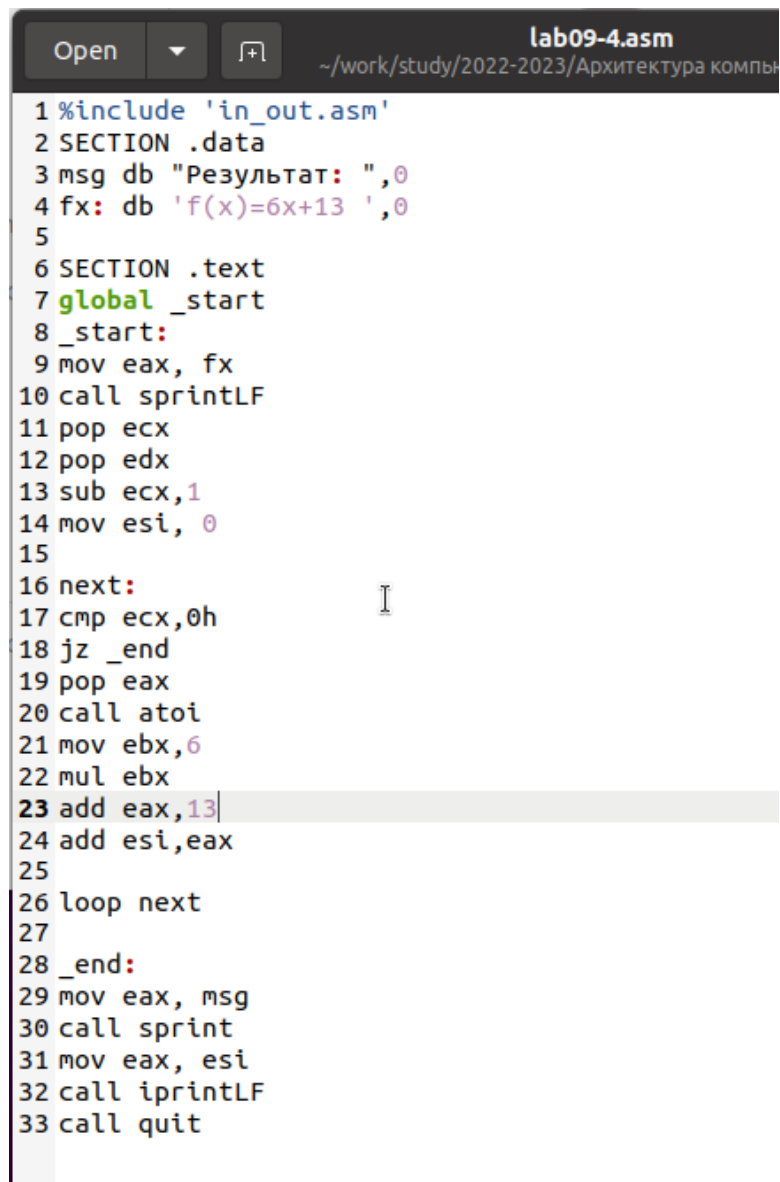
```

Рис. 2.12: Работа программы lab9-3.asm

8. Напишите программу, которая находит сумму значений функции  $f(x)$  для  $x = x_1, x_2, \dots, x_n$ , т.е. программа должна выводить значение  $f(x_1) + f(x_2) + \dots + f(x_n)$ . Значения  $x$  передаются как аргументы. Вид функции  $f(x)$  выбрать из таблицы 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. [2.13], [2.14])

для варианта 15  $f(x) = 6x + 13$





```
lab09-4.asm
~/work/study/2022-2023/Архитектура компь

1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 fx: db 'f(x)=6x+13 ',0
5
6 SECTION .text
7 global _start
8 _start:
9 mov eax, fx
10 call sprintf
11 pop ecx
12 pop edx
13 sub ecx,1
14 mov esi, 0
15
16 next:
17 cmp ecx,0h
18 jz _end
19 pop eax
20 call atoi
21 mov ebx,6
22 mul ebx
23 add eax,13
24 add esi,eax
25
26 loop next
27
28 _end:
29 mov eax, msg
30 call sprintf
31 mov eax, esi
32 call iprintLF
33 call quit
```

Рис. 2.13: Файл lab9-4.asm

```
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура комп  
labs/lab09$ nasm -f elf lab09-4.asm  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура комп  
labs/lab09$ ld -m elf_i386 -o lab09-4 lab09-4.o  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура комп  
labs/lab09$ ./lab09-4 1 2 3 9 7 1 2 3  
f(x)=6x+13  
Результат: 272  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура комп  
labs/lab09$ ./lab09-4 1  
f(x)=6x+13  
Результат: 19  
viktoronvudive@VirtualBox:~/work/study/2022-2023/Архитектура комп  
labs/lab09$
```

Рис. 2.14: Работа программы lab9-4.asm

## 3 Выводы

Освоили работы со стеком, циклом и аргументами на ассемблере `naasm`.