

Создание файла lab8-1.asm, компиляция и выполнение кода

```
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ gedit lab8-1.asm
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ nasm -f elf lab8-1.asm
ld -m elf_i386 lab8-1.o -o lab8-1
./lab8-1
Введите N: 5
5
4
3
2
1
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ 
```

Изменение с изменением ECX

```
29 label:
30     sub ecx, 1
31     mov [N],ecx
32     mov eax,[N]
33     call iprintLF
34     loop label
35     call quit
Matlab ▾ Largeur des tabulations : 8
```

Использование PUSH/POP для сохранения ECX

```
29 label:
30     push ecx      ; добавление значения ecx в стек
31     sub ecx, 1    ; modifier ecx
32     mov [N], ecx
33     mov eax, [N]
34     call iprintLF
35     pop ecx       ; извлечение значения ecx из стека
36     loop label    ; loop utilise la valeur ORIGINALE de ecx
37
38 call quit
```

ОБРАБОТКА АРГУМЕНТОВ, Компиляция и тестирование

```
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ touch lab8-2.asm
gedit lab8-2.asm
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ nasm -f elf lab8-2.asm
ld -m elf_i386 lab8-2.o -o lab8-2
./lab8-2 аргумент1 аргумент2 'аргумент 3'
аргумент1
аргумент2
аргумент 3
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ 
```

```

1 %include 'in_out.asm'
2
3 SECTION .text
4 global _start
5
6 _start:
7     pop ecx      ; Извлекаем количество аргументов
8     pop edx      ; Извлекаем имя программы
9     sub ecx, 1   ; Уменьшаем количество аргументов
10
11 next:
12     cmp ecx, 0  ; проверяем, есть ли еще аргументы
13     jz _end      ; если аргументов нет выходим
14     pop eax      ; извлекаем аргумент из стека
15     call sprintLF ; печать аргумента
16     loop next    ; переход к следующему
17
18 _end:
19     call quit

```

Программа суммы (lab8-3.asm) и тестирование

```

lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ touch lab8-3.asm
gedit lab8-3.asm
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ nasm -f elf lab8-3.asm
ld -m elf_i386 lab8-3.o -o lab8-3
./lab8-3 12 13 7 10 5
Результат: 47
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ 

```

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8
9 _start:
10    pop ecx      ; Извлекаем количество аргументов
11    pop edx      ; Извлекаем имя программы
12    sub ecx, 1   ; Уменьшаем количество аргументов
13    mov esi, 0    ; Используем `esi` для хранения суммы
14
15 next:
16    cmp ecx, 0  ; проверяем, есть ли еще аргументы
17    jz _end      ; если аргументов нет выходим
18    pop eax      ; извлекаем следующий аргумент
19    call atoi    ; преобразуем символ в число
20    add esi, eax ; добавляем к сумме
21    loop next    ; переход к следующему аргументу
22
23 _end:
24    mov eax, msg  ; вывод сообщения "Результат: "
25    call sprint
26    mov eax, esi  ; записываем сумму в регистр `eax`
27    call iprintLF ; печать результата
28    call quit      ; завершение программы

```

Модификация для продукта: Изменения в lab8-3.asm

```

lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ gedit lab8-3.asm
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ nasm -f elf lab8-3.asm
ld -m elf_i386 lab8-3.o -o lab8-3
./lab8-3 2 3 4
Результат: 24
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ 

```

САМОСТОЯТЕЛЬНАЯ РАБОТА

**Программа суммы функции f(x) Выберите функцию из таблицы 8.1 (например:
 $f(x) = 2x + 15)$**

```
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ touch lab8-4.asm
gedit lab8-4.asm
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ nasm -f elf lab8-4.asm
ld -m elf_i386 lab8-4.o -o lab8-4
./lab8-4 1 2 3 4
Функция: f(x)=2x+15
Результат: 80
lissombayem@fedora:~/work/study_2025-2026_ACH-PC/labs/lab08$ □
```