Шаблон отчёта по лабораторной работе

Дицсиплина:Архитектура компьютеров

Хатамов Эзиз

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

1. Реализация циклов в NASM
2. Обработка аргументов командной строки

# 3 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. На рис. 8.1 показана схема организации стека в процессоре. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается. Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop).

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Она позволяет организовать безусловный цикл, типичная структура которого имеет следующий вид: mov ecx, 100 ; Количество проходов NextStep: … … ; тело цикла … loop NextStep ; Повторить ecx раз от метки NextStep Иструкция loop выполняется в два этапа. Сначала 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.

# 4 Выполнение лабораторной работы

##Реализация циклов в NASM Для начала я создал каталог для программам лабораторной работы № 8, перешёл в него и создайл файл lab8-1.asm:

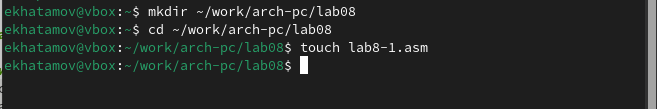


Рис. 1: Создания каталога и файла

После этого как я делал в предидущих лабораторных работ я скопирую файл in\_out.asm в новый созданный каталог

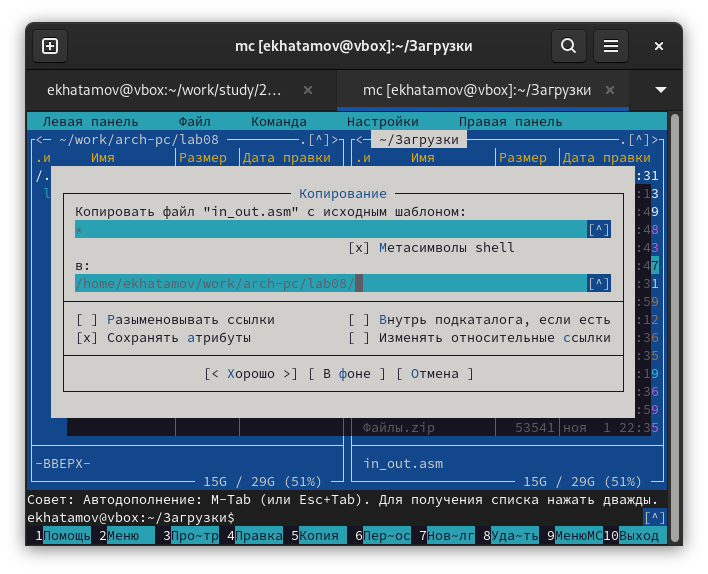


Рис. 2: Копирования файла in\_out.asm

После этого я открыл новый созданный файл и написал туда код которая выводит значение регистра ecx.

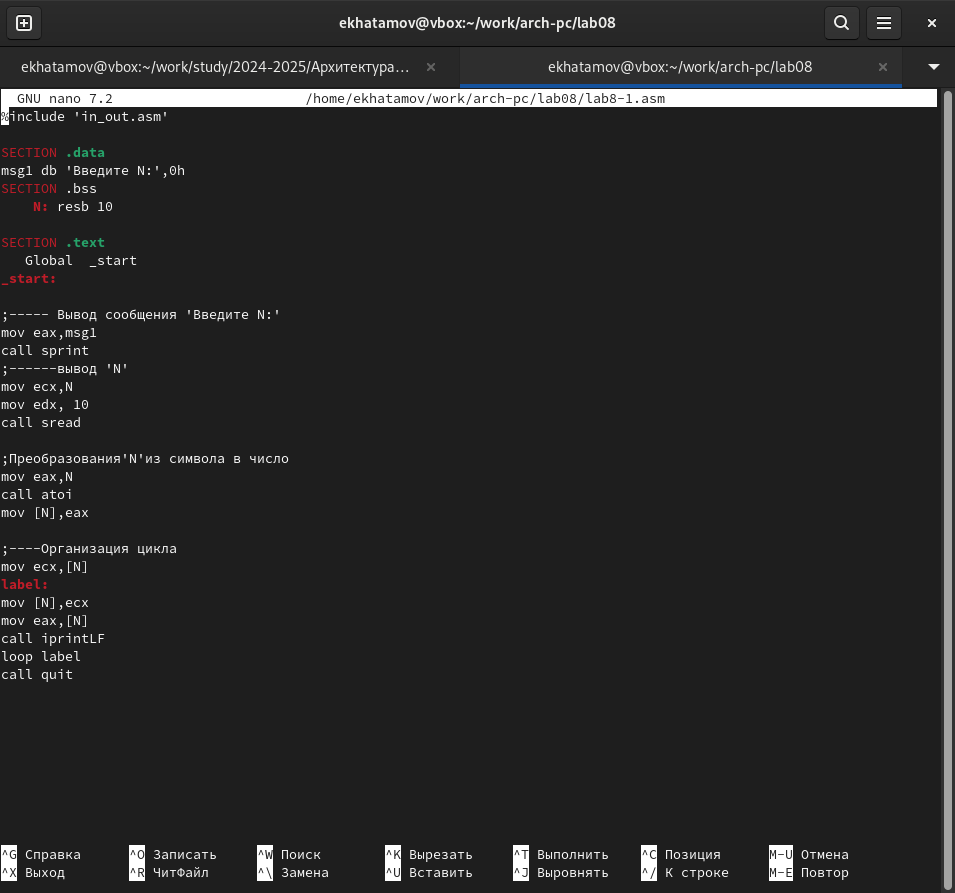


Рис. 3: Введения кода в файл

Потом я создал выполняемый файл и запустил его

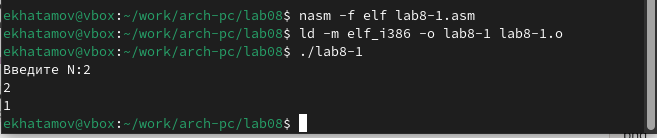


Рис. 4: Создания выполняемого файла lab8-1.asm

Потом я создал копию файла c именем lab8-1-1.asm

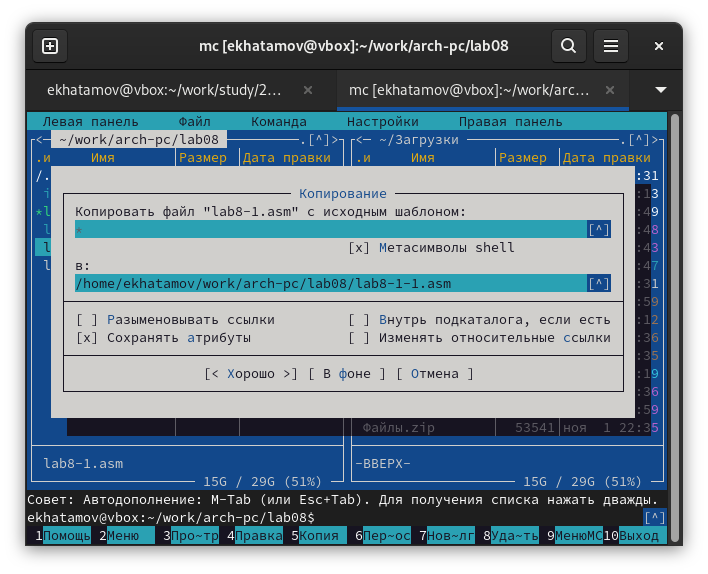


Рис. 5: Создания копии файла lab8-1.asm

Я открыл новый созданный файл и в нем измененил текст так использование регистра ecx в теле цилка loop может привести к некорректной работе программы

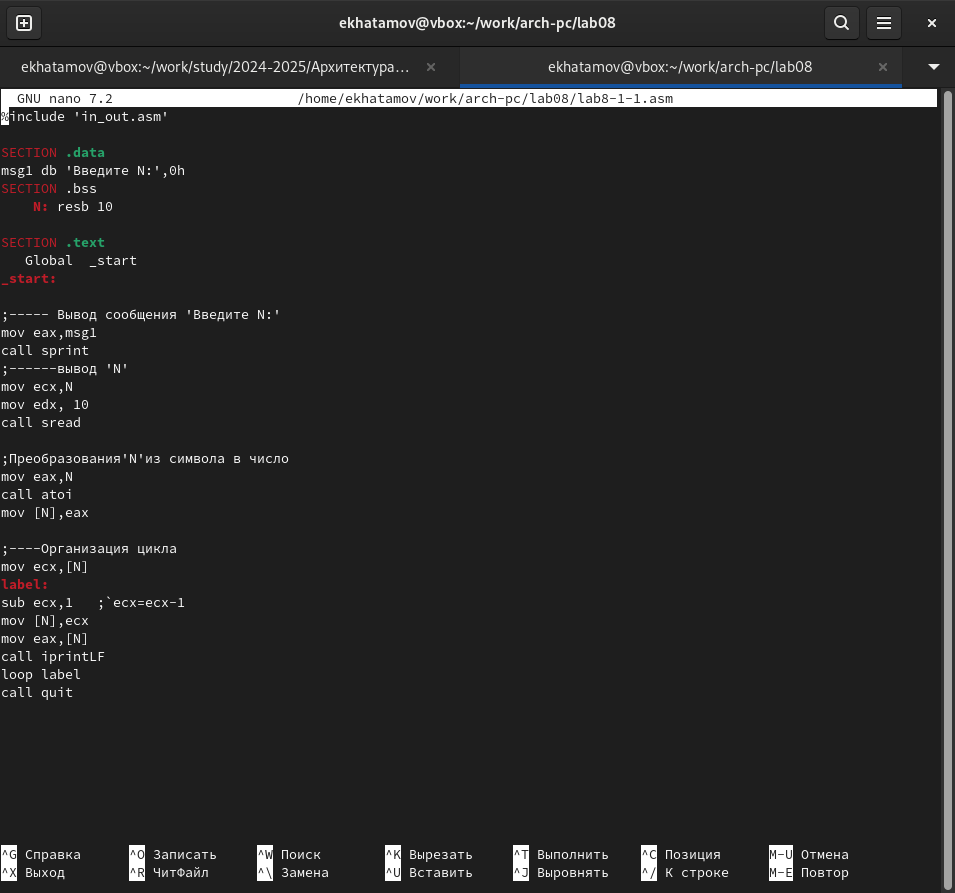


Рис. 6: Исправления текста в файле

Потом я создал выполняемый файл и запустил его

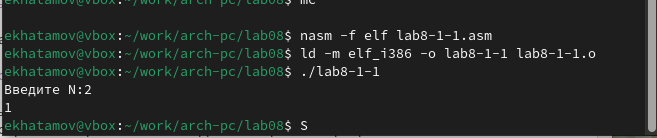


Рис. 7: Создания выполняемого файла lab8-1-1.asm

Потом я создал новую копию файла c именем lab8-1-2.asm

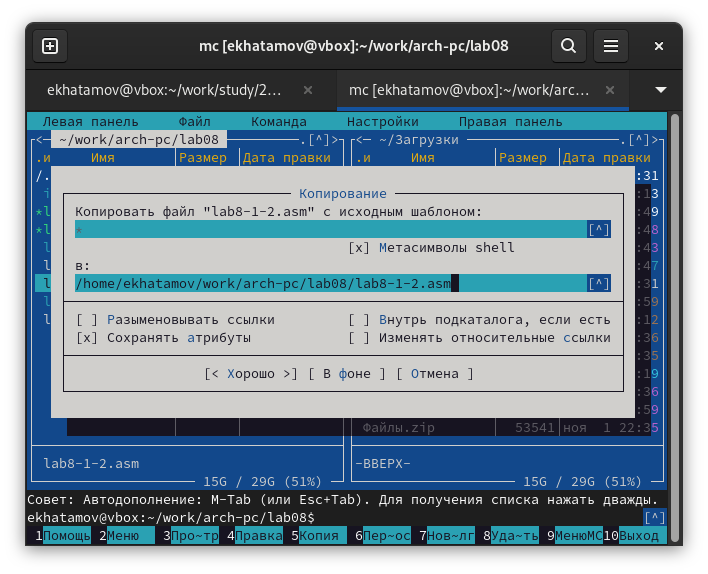


Рис. 8: Создания второй копии файла lab8-1.asm

После этого для использования регистра ecx в цикле и сохранения корректности работы программы использовал стек. Внес изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:

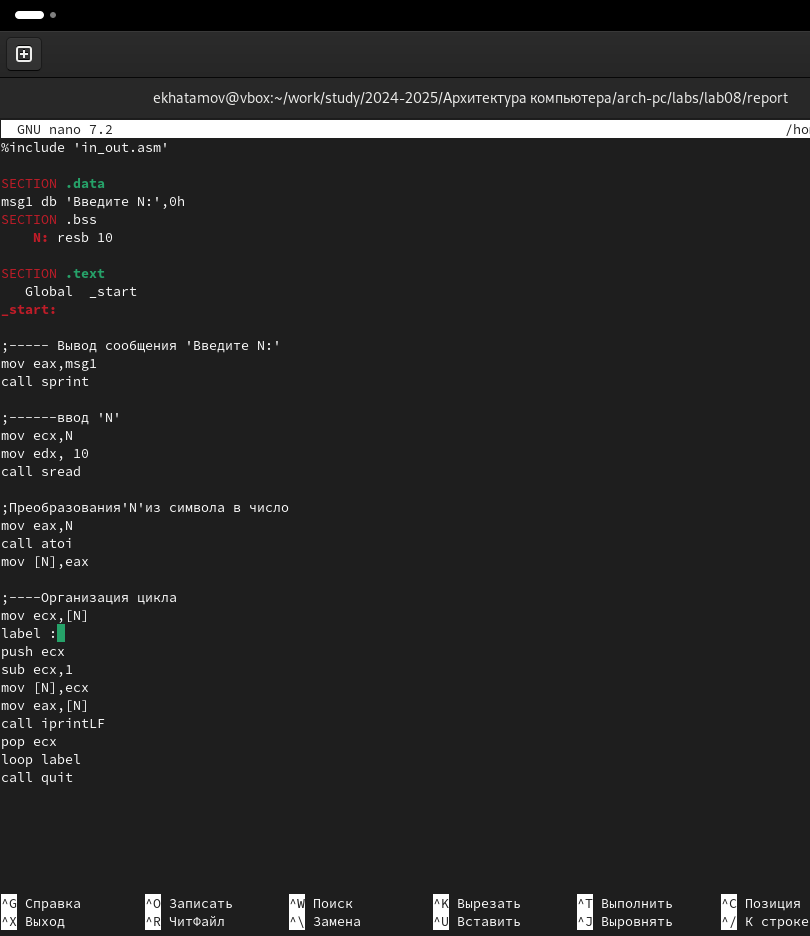


Рис. 9: Исправления текста в файле lab8-1-2.asm

Потом создал исполняемый файл и запустил его

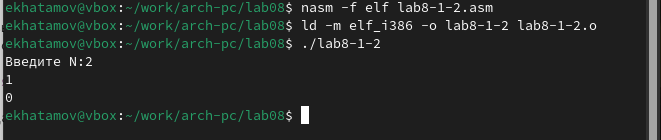


Рис. 10: Создания выполняемого файла lab8-1-2.asm

## 4.1 Обработка аргументов командной строки

Для начало я создал создал новый файл для написание программы которая выводит на экран аргументы командной строки.

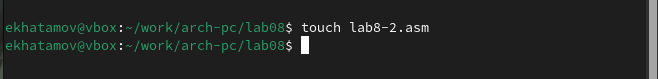


Рис. 11: Создания файла

После этого я написал программу в созданный файл

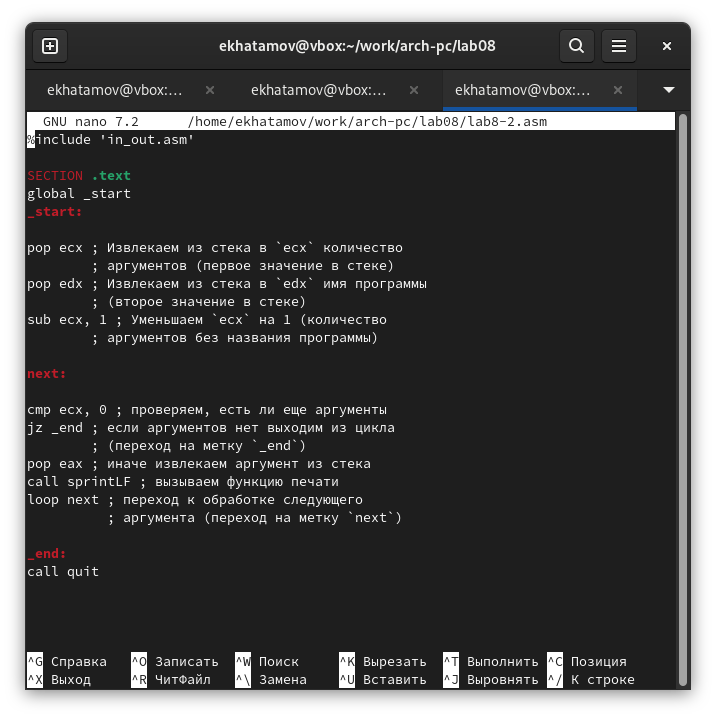


Рис. 12: Программа выводящая на экран аргументы командной строки

Создал исполняемый файл и запустил его, указав аргументы

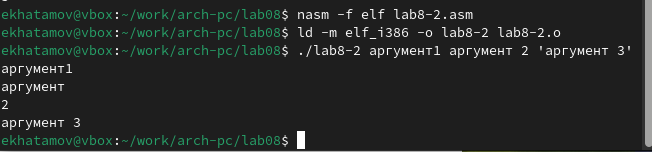


Рис. 13: Создания исполняемого файла и запуск его с указаннами аргументами

Потом я скопировал файл и изменил программу, чтобы она выводила произведение введенных чисел.

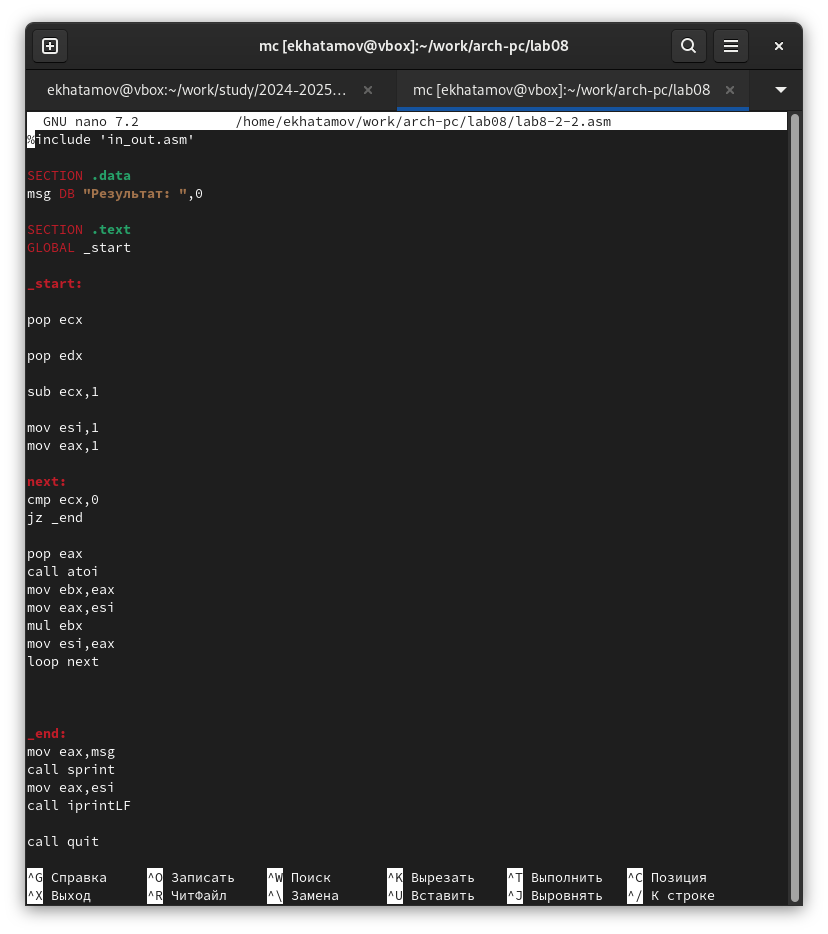


Рис. 14: Скопирования файла lab8-2-1

После этого я создал исолняемый файл и запустил его.

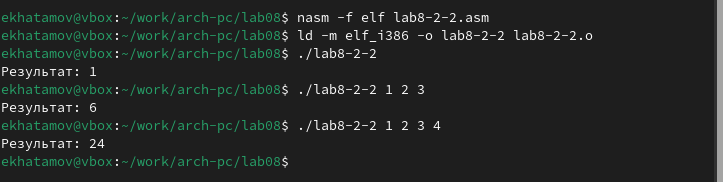


Рис. 15: программа которая выводит сумму всех решений примера

# 5 Задание для самостоятельной работы

Для начала я создал файл для самостоятельной работы

Рис. 16: Создания файла для самостоятельной работы

Рис. 16: Создания файла для самостоятельной работы

Я написал программу, которая выводит сумму всех решений примера. В лабораторной работе №7, я получил 1 вариант, поэтому я писал программу для первого варианта. Введенные числа я придумал сам, и посчитал их, чтобы проверить работу программы.

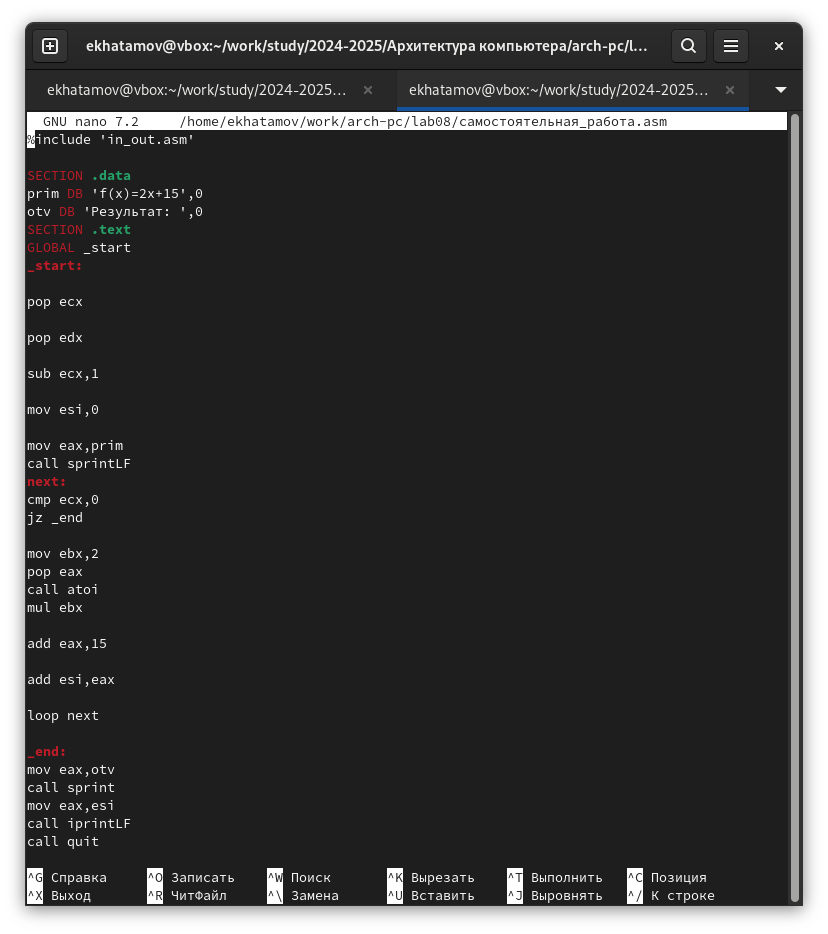


Рис. 17: программа которая выводит сумму всех решений примера

Потом я создал исполняемый файл и запустил его, с аргументами. Проверил файл все ли правильно сделано

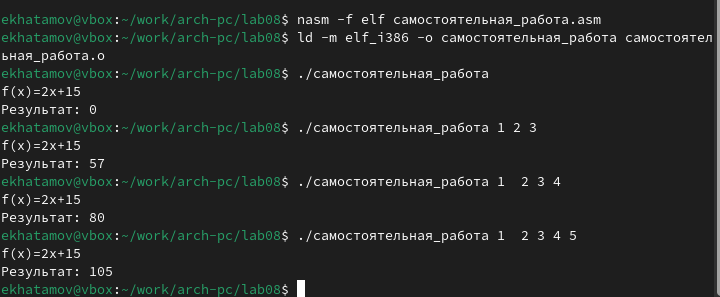


Рис. 18: Результаты работы

# 6 Выводы

Я приобрел навыки написания программы с использованием цикла.

# Список литературы

https://esystem.rudn.ru/pluginfile.php/2089548/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F %D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0 %E2%84%968. %D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5 %D1%86%D0%B8%D0%BA%D0%BB%D0%B0. %D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0 %D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2 %D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9 %D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8..pdf :::