Отчет по лабораторной работе №7

Арифметические операции в NASM

Виктор Максимович Кадров

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

1. Символьные и численные данные в NASM.
2. Выполнение арифметических операций в NASM.
3. Ответы на вопросы.
4. Выполнение заданий для самостоятельной работы.

# 3 Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинноориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр. Программы, написанные на языке ассемблера, не уступают в качестве и скорости программам, написанным на машинном языке, так как транслятор просто переводит мнемонические обозначения команд в последовательности бит (нулей и единиц). Используемые мнемоники обычно одинаковы для всех процессоров одной архитектуры или семейства архитектур (среди широко известных — мнемоники процессоров и контроллеров x86, ARM, SPARC, PowerPC,M68k). Таким образом для каждой архитектуры существует свой ассемблер и, соответственно, свой язык ассемблера.

# 4 Выполнение лабораторной работы

## 4.1 Символьные и численные данные в NASM.

Создаем папку lab07 и файл lab07-1.asm в ней (рис. 1). Пишем код программы, выводящей значение регистра (рис. 2). Выполняем команду и проверяем вывод (рис. 3). Изменяем код программы так, чтобы в регистр записывались числа, а не символы. (рис. 4). Выполняем ее (рис. 5). В соответствии с таблцей кодирования ASCII, код 10 соответствует символу переноса каретки, поэтому на экране он не отображается. Записываем в файл lab07-2.asm код программы (рис. 6), выводящей значение регистра без перевода в ASCII, и выполняем ее (рис. 7). Изменяем код аналогично lab-07-1.asm (рис. 8) и выполняем (рис. 9). В результате в консоль выводится значение регистра без перевода в ASCII. После этого меняем функцию iprintLF на iprint. Их откличие заключается в том, что первая выводит строку и добавляет символ переноса каретки.

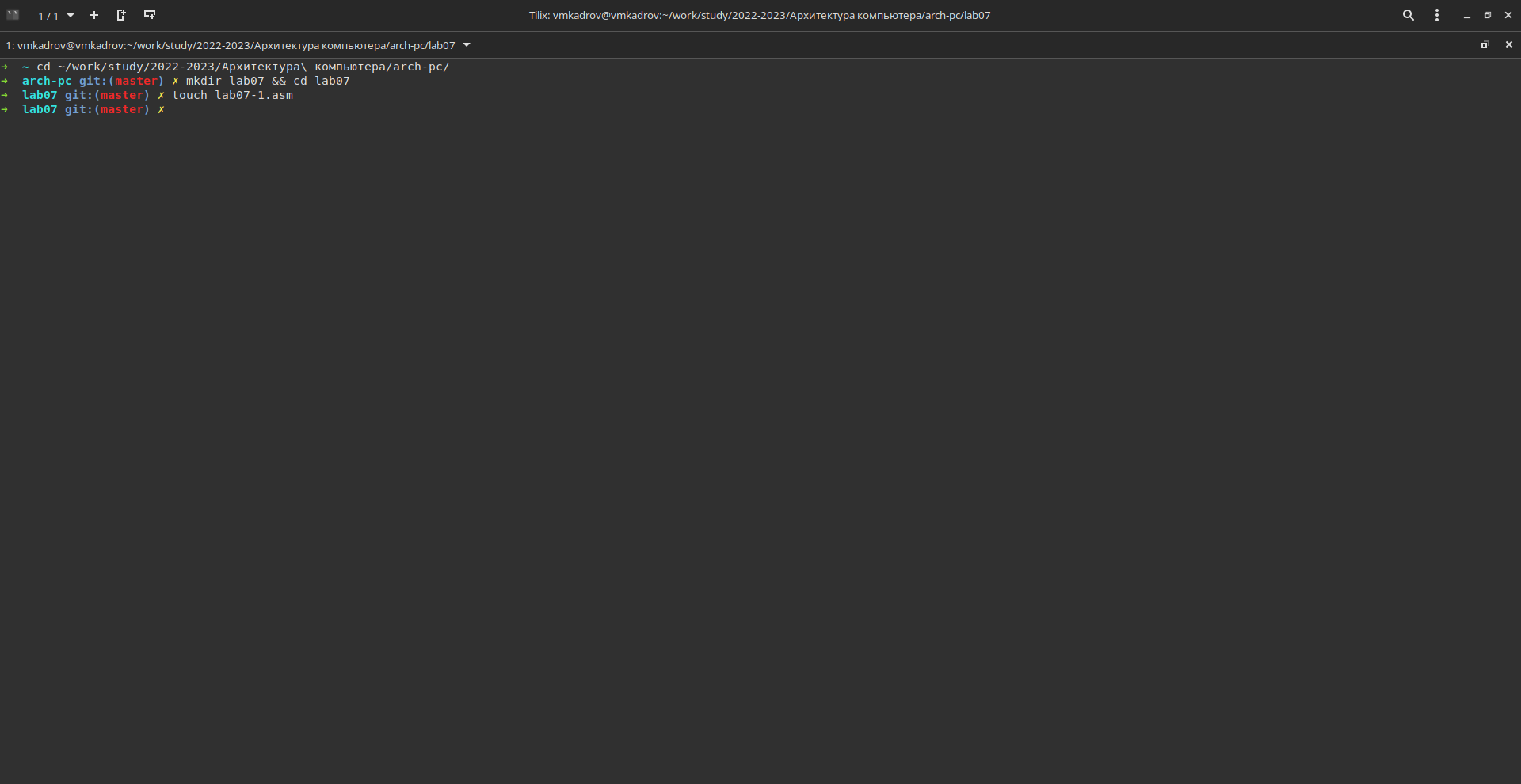


Рис. 1: Создание папки и файла

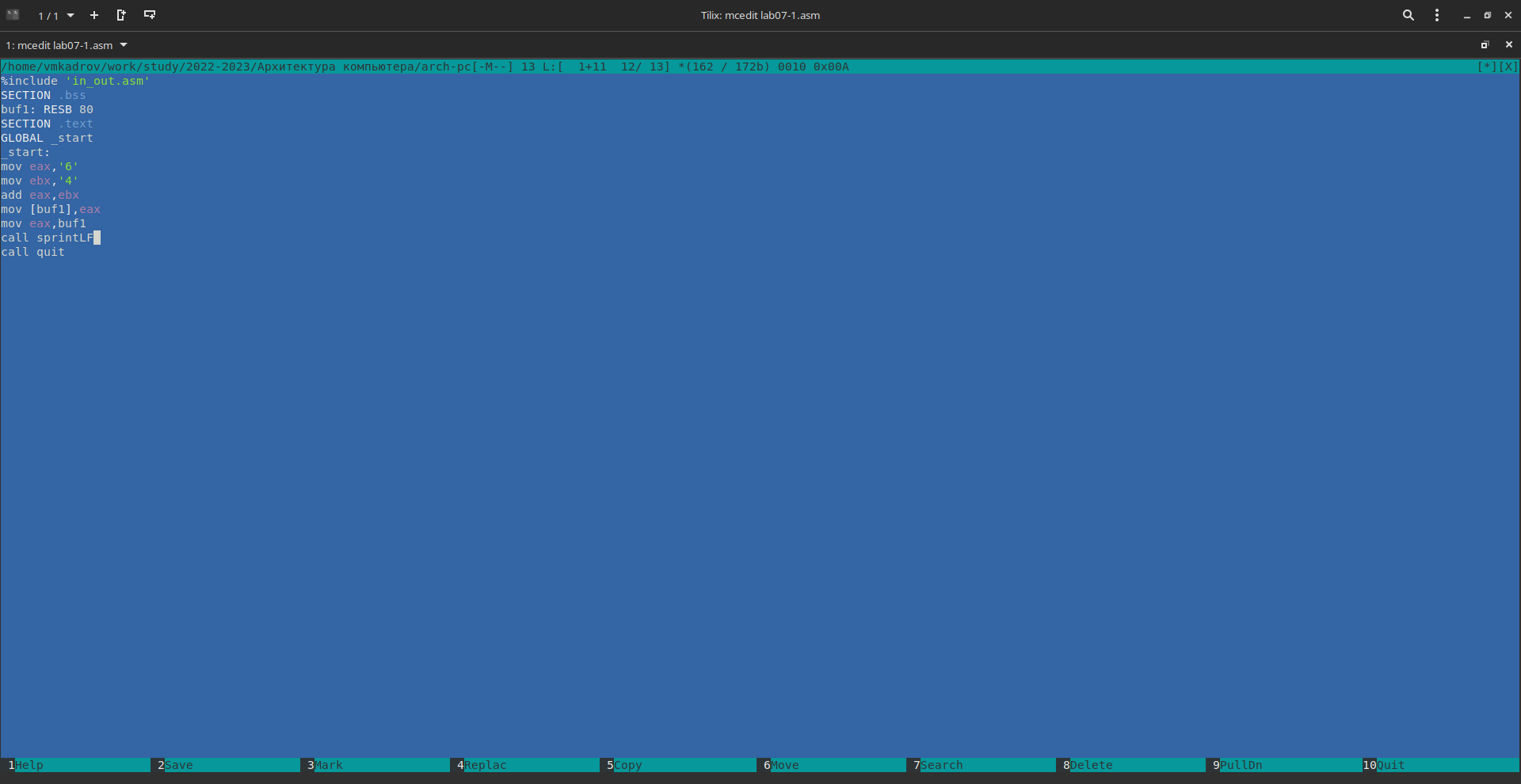


Рис. 2: Код lab07-1.asm

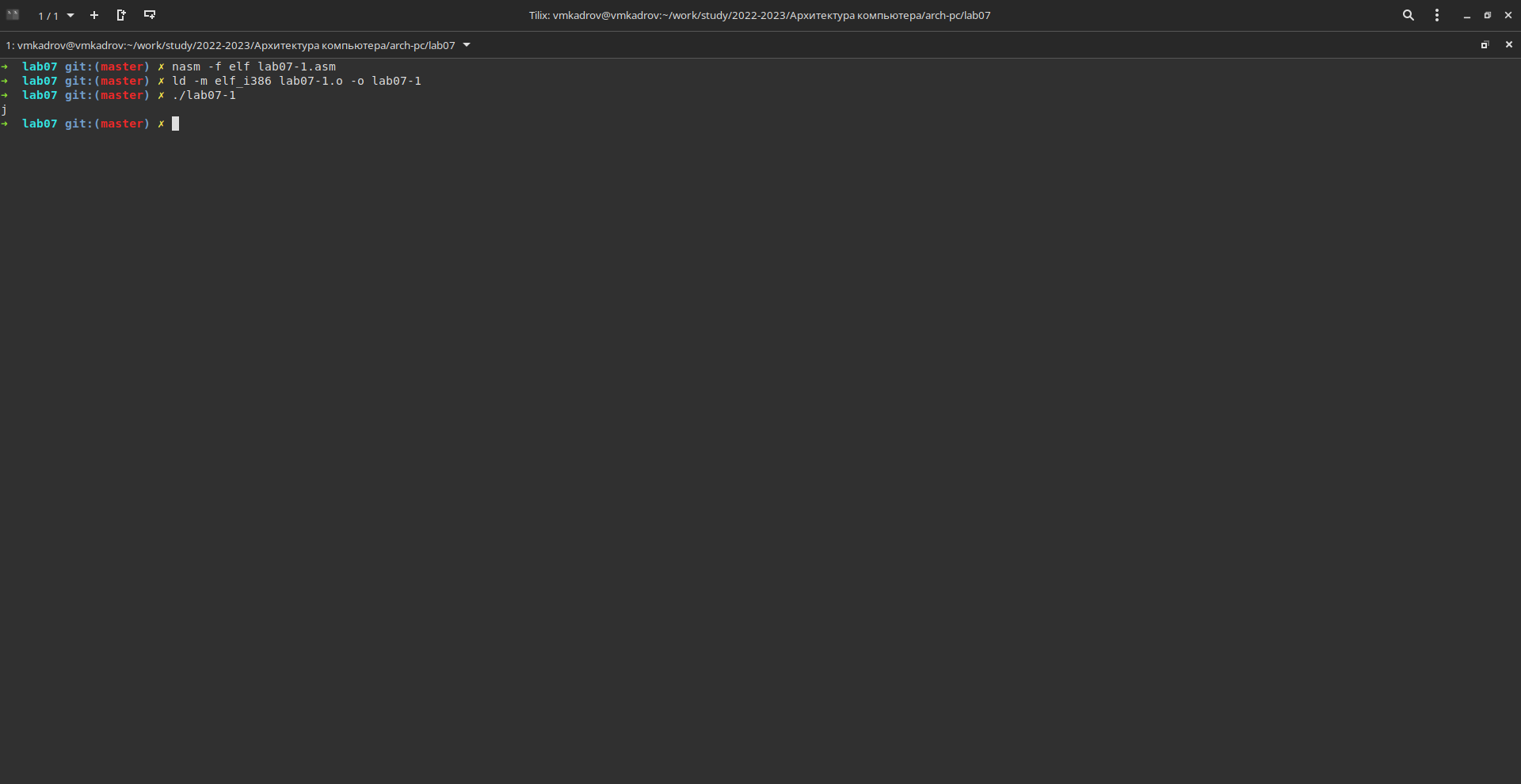


Рис. 3: Вывод программы

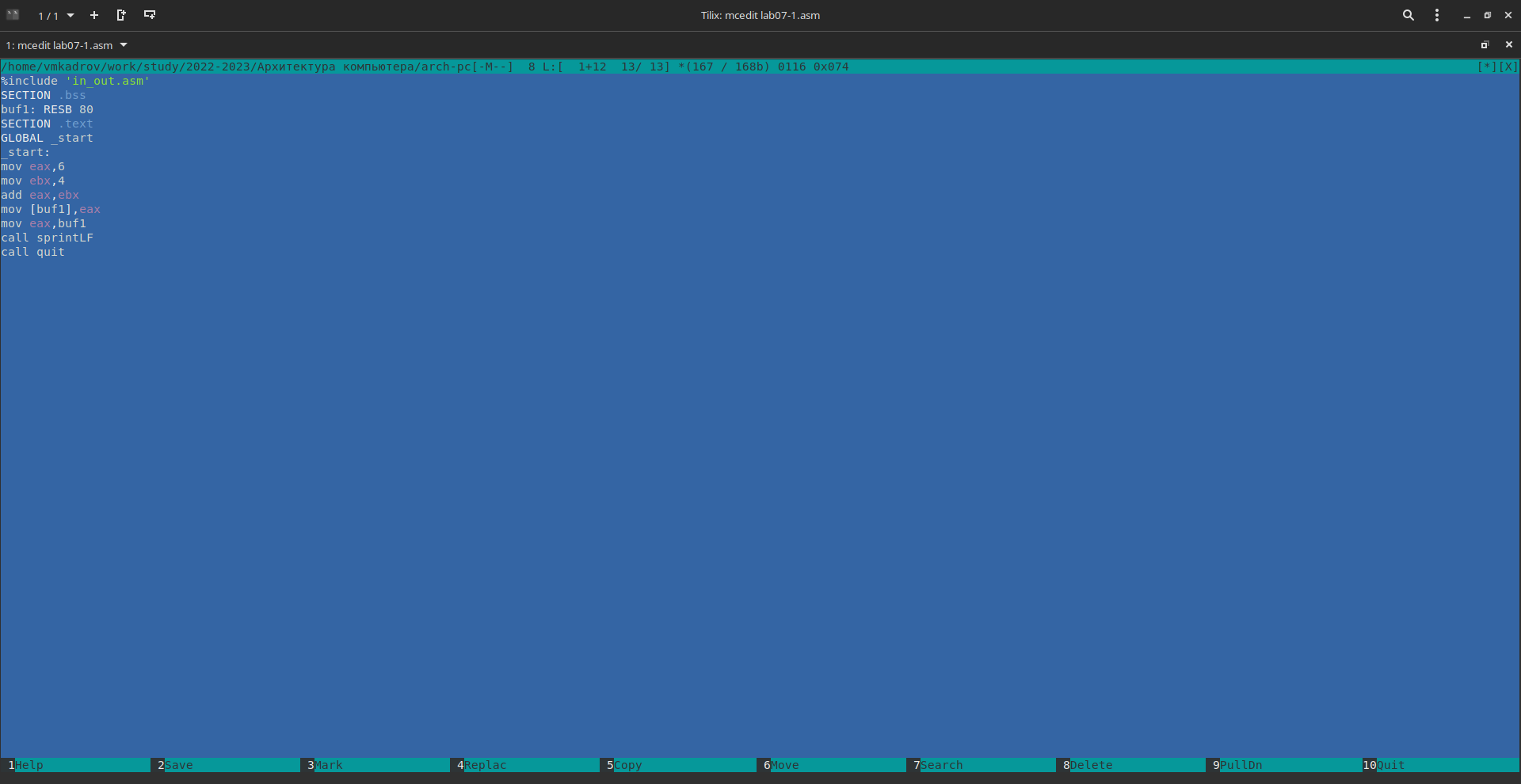


Рис. 4: Измененный код

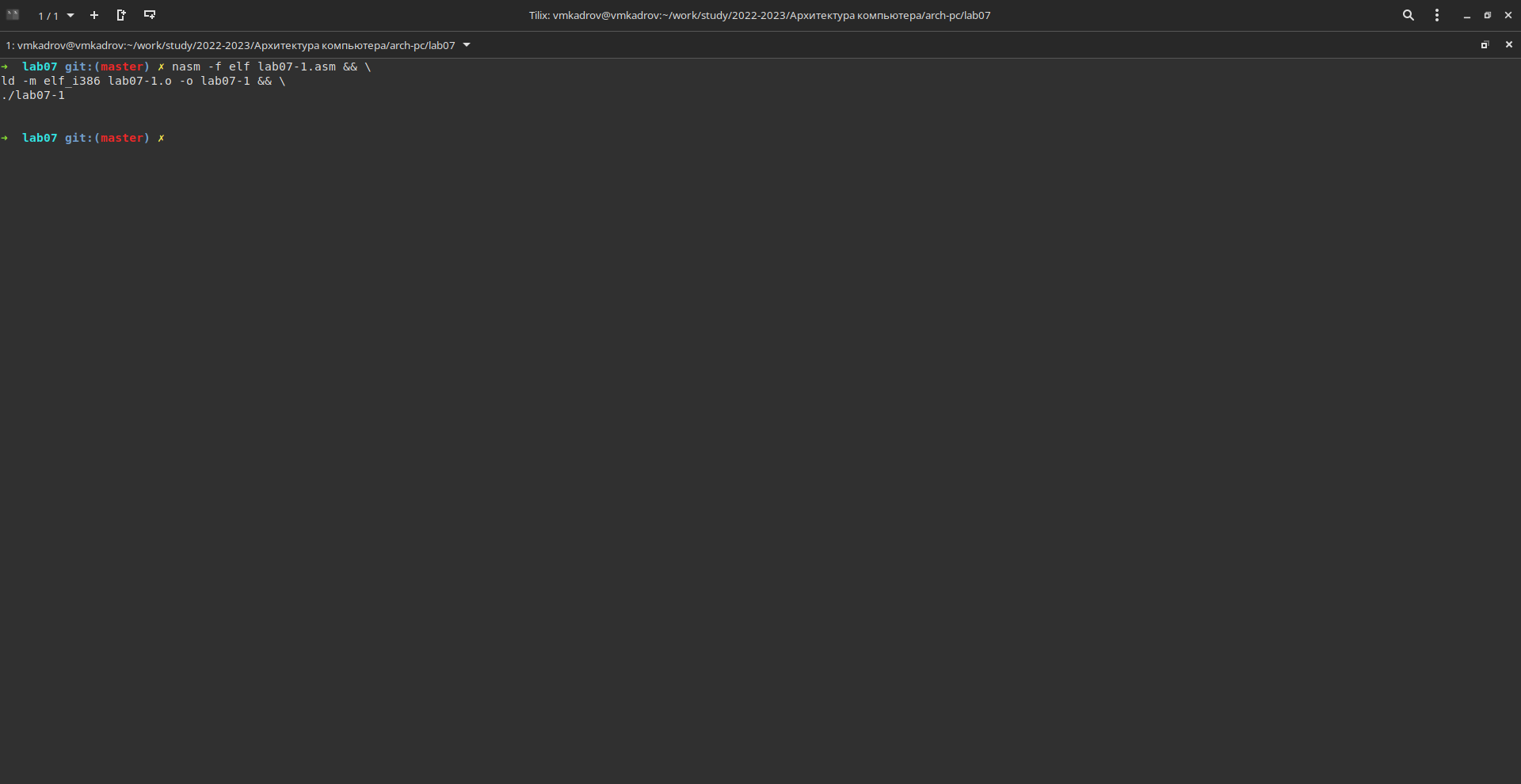


Рис. 5: Выплнение новой программы

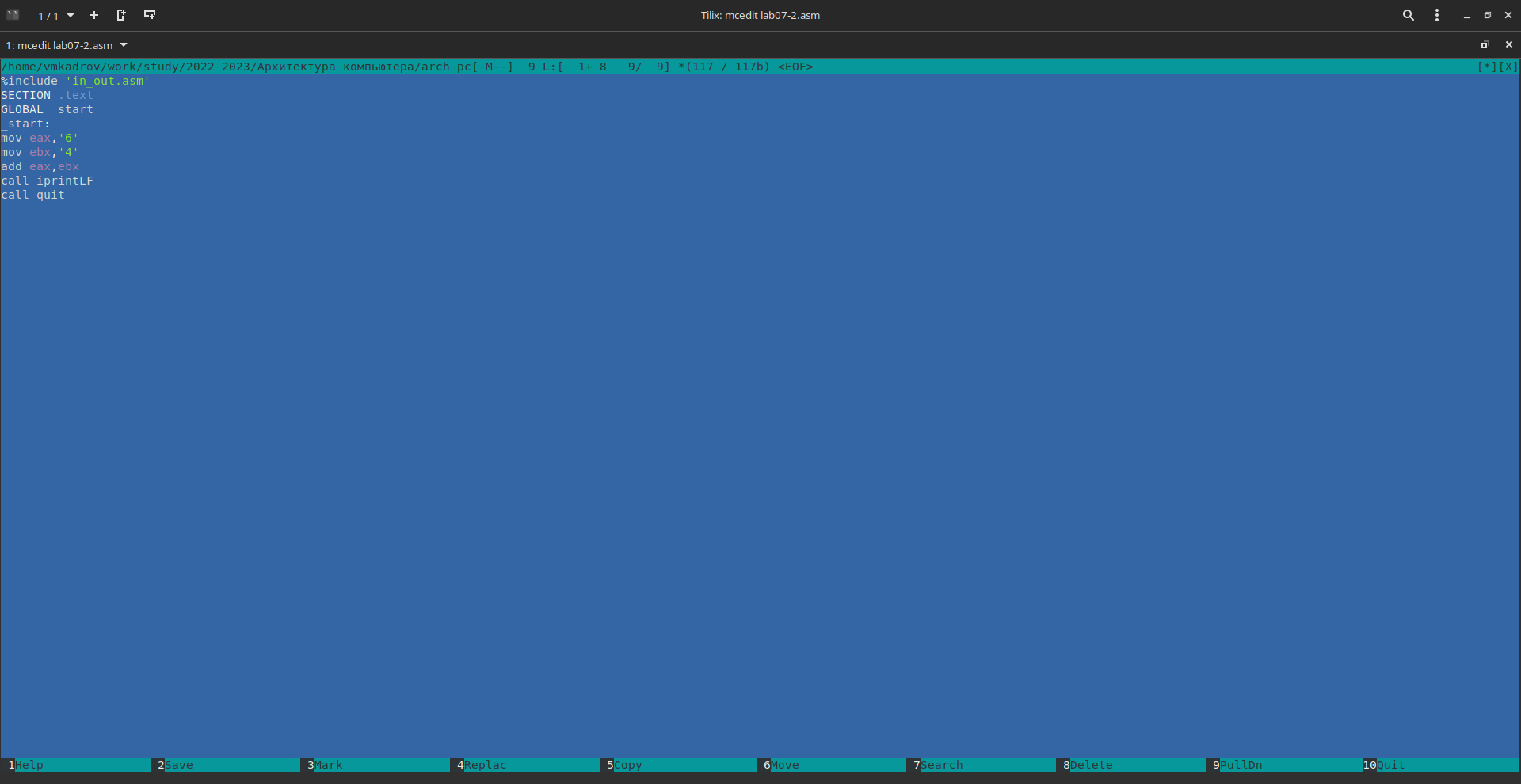


Рис. 6: Код ab07-2.asm

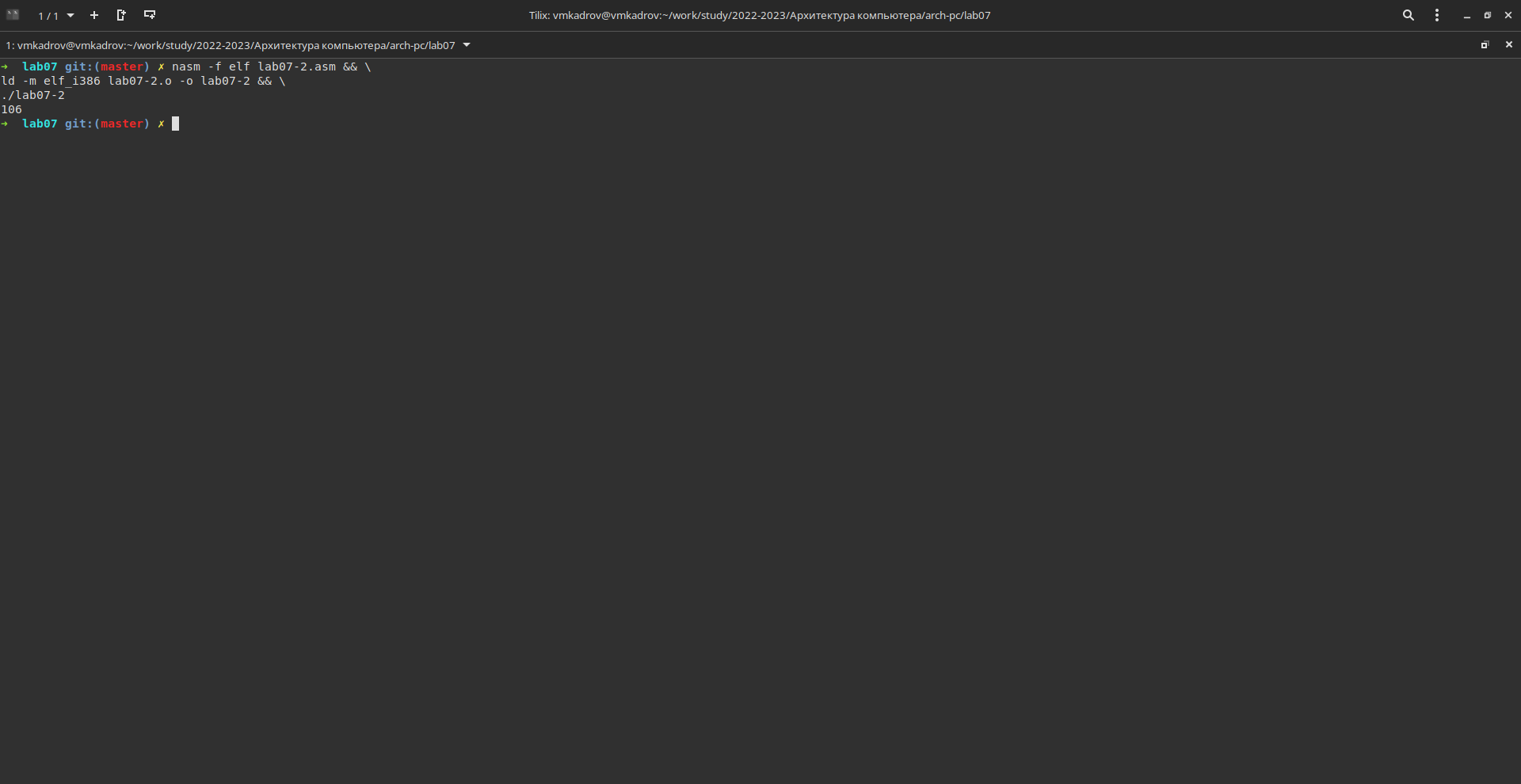


Рис. 7: Вывод программы

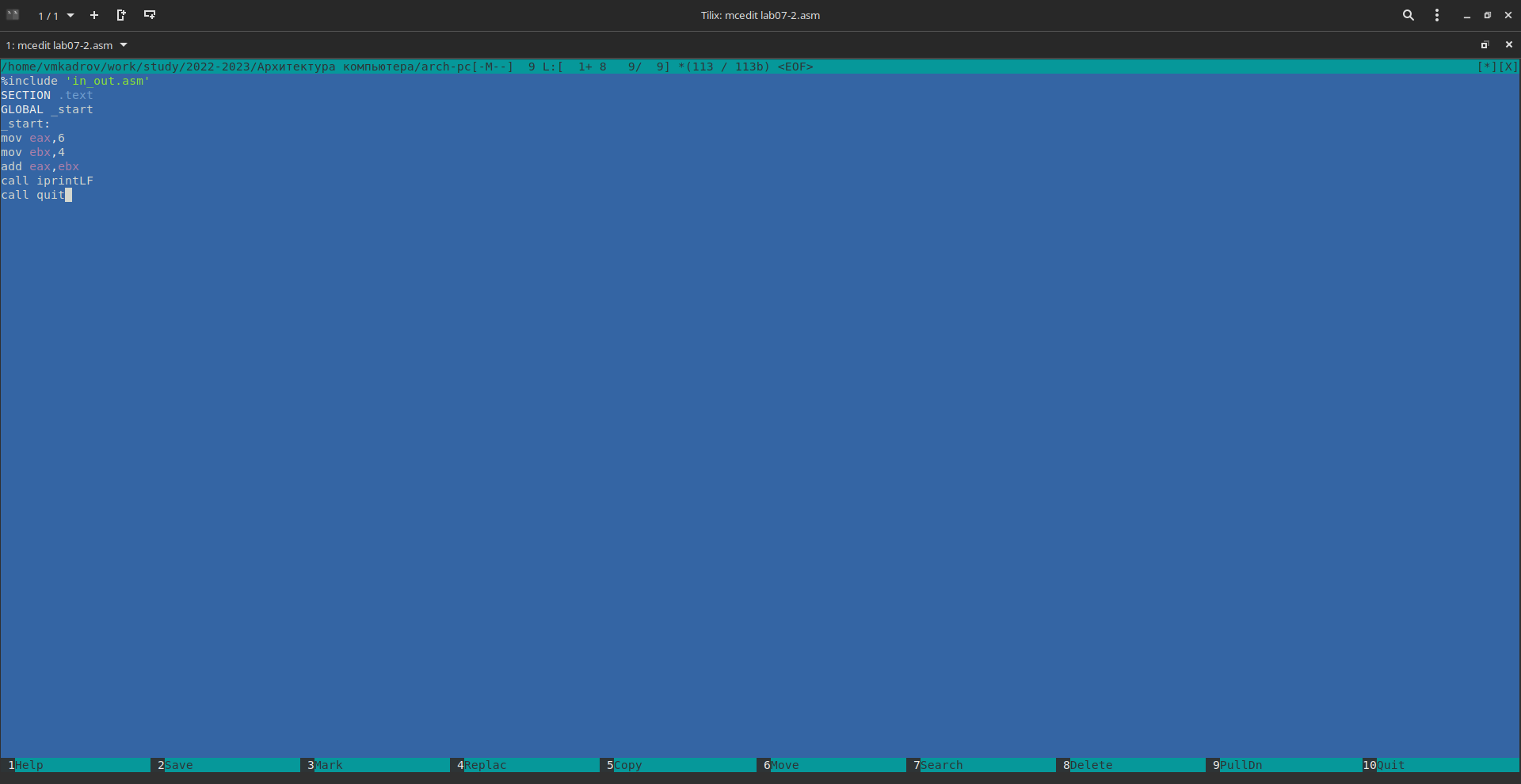


Рис. 8: Измененный код

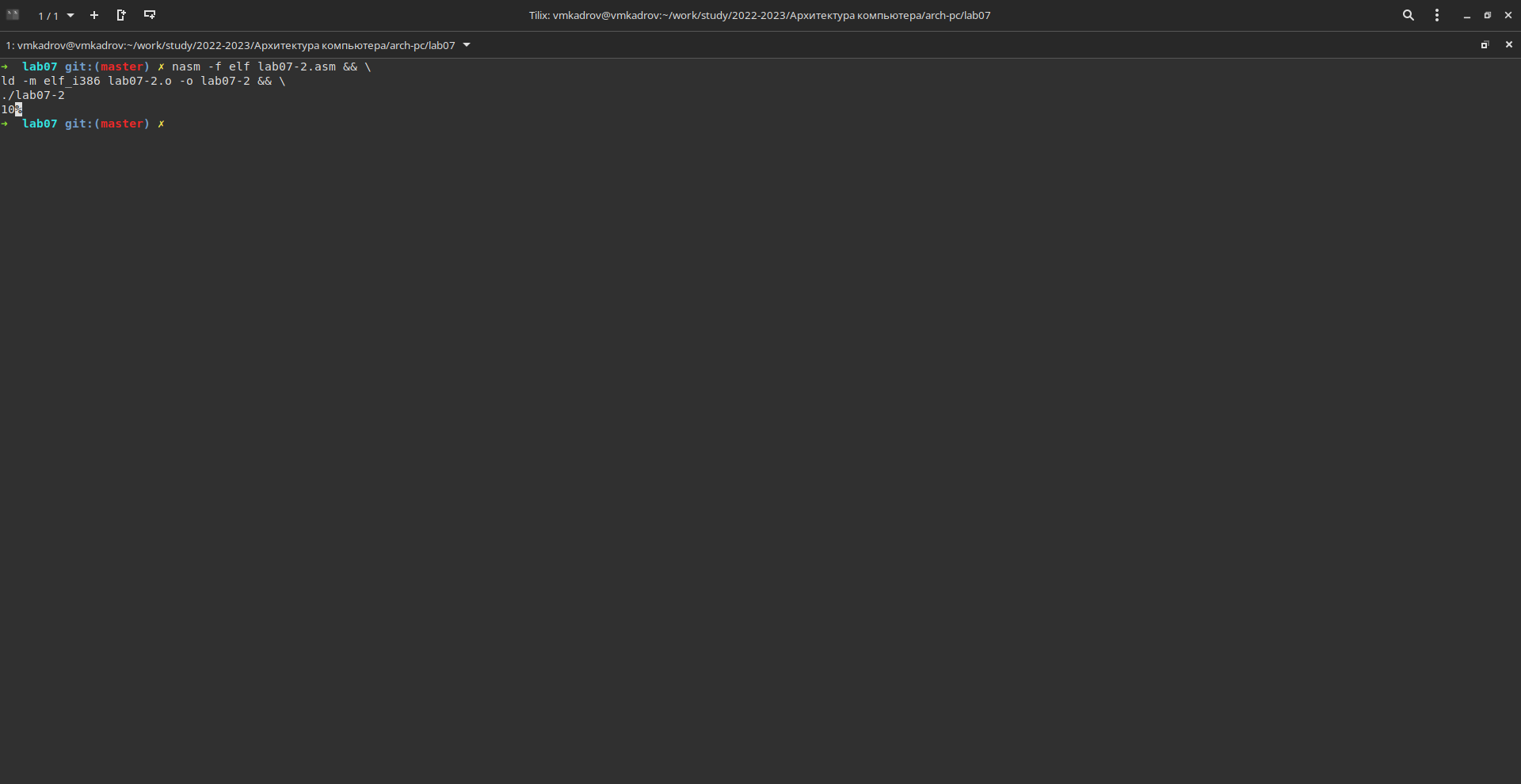


Рис. 9: Вывод lab07-2

## 4.2 Выполнение арифметических операций в NASM.

Создаем файл lab07-3.asm и пишем туда код программы (рис. 10), считающей результат арифметического выражения. Выполняем ее и проверяем корректность вывода (рис. 11). Изменяем код так, чтобы программы считала другое выражение (рис. 12). Выпоняем его (рис. 13). Создаем файл variant.asm и пишем туда код программы, расcчитывающей номер варианта (рис. 14) и выполняем ее (рис. 15).

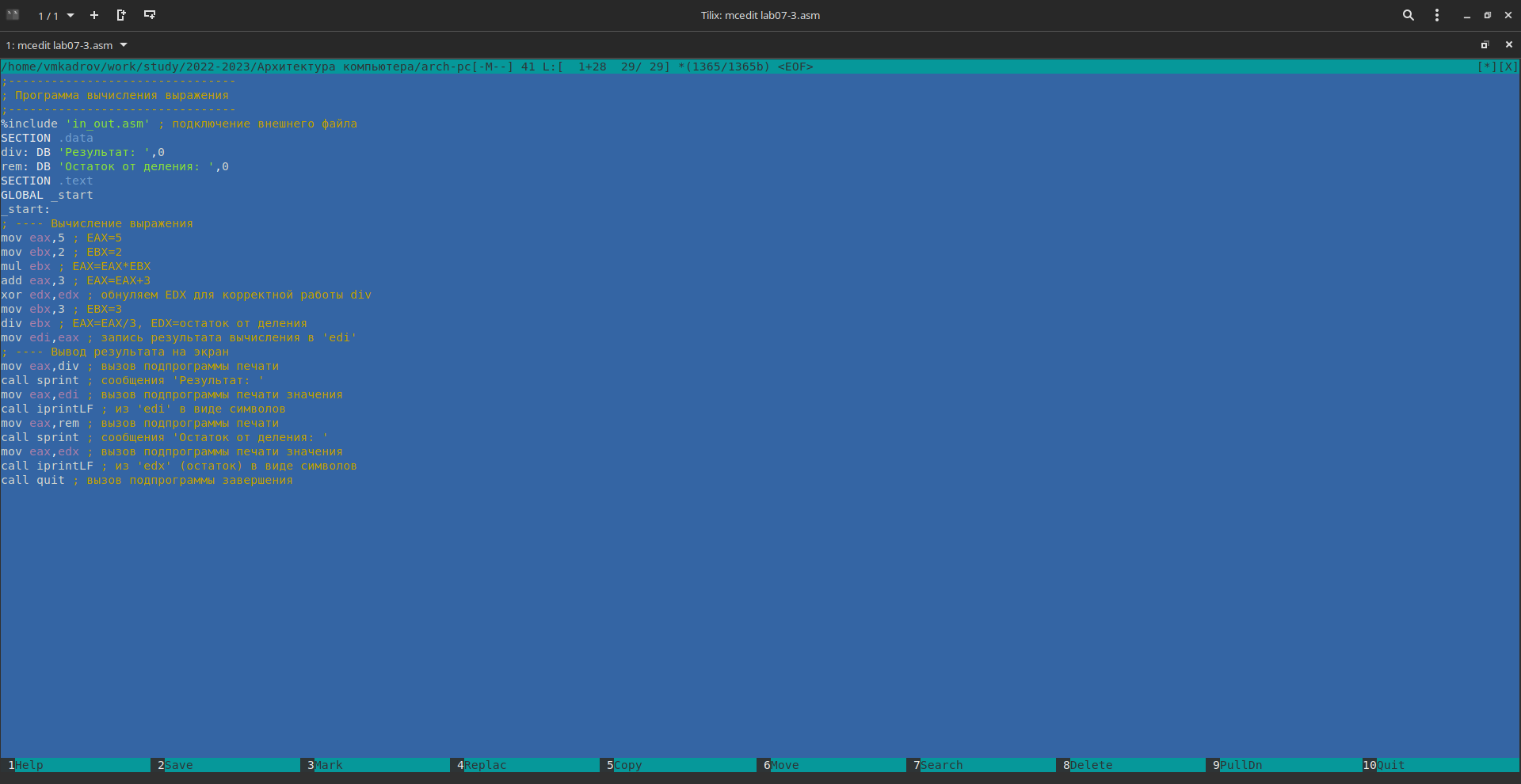


Рис. 10: Код lab07-3.asm

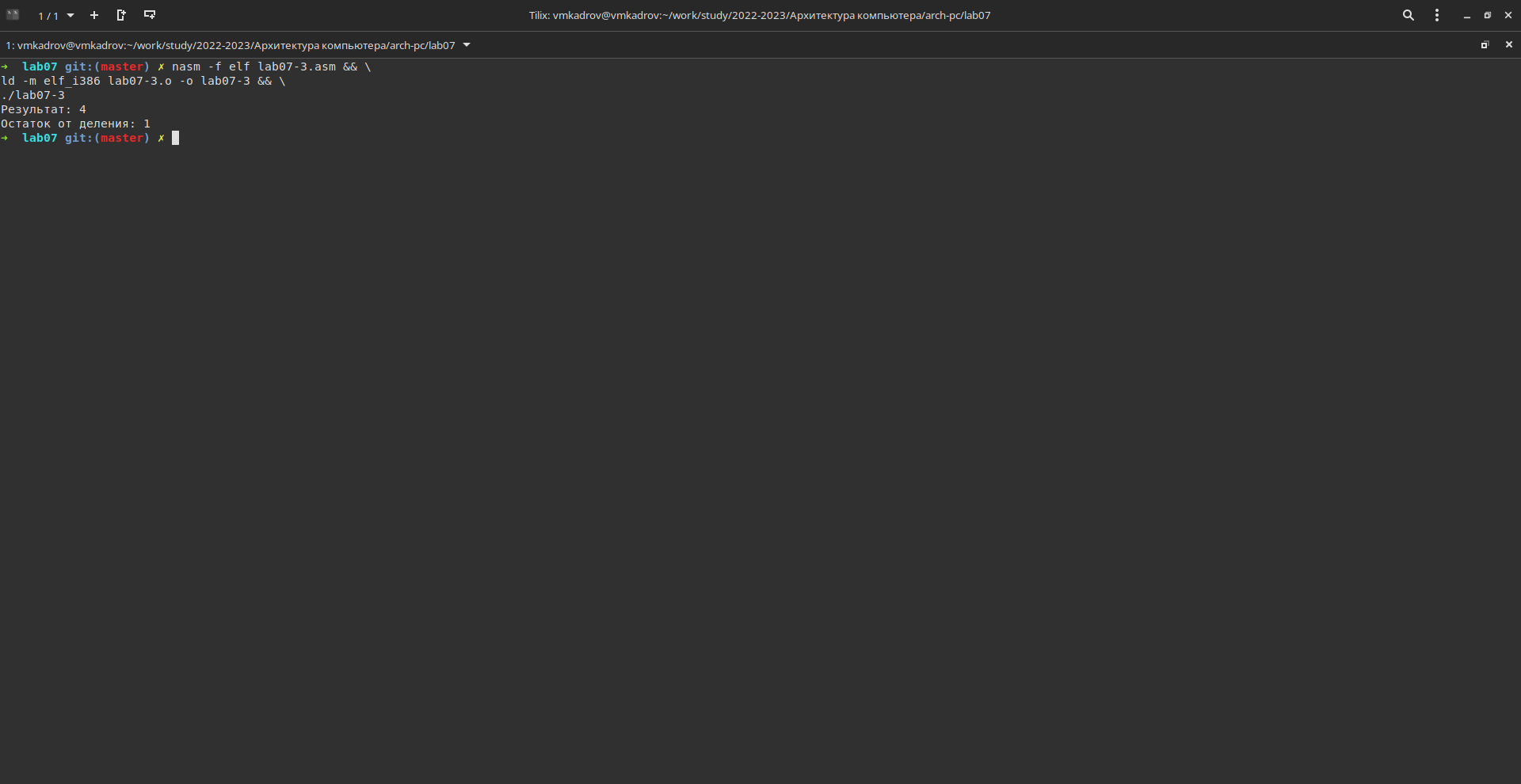


Рис. 11: Результат выполнения программы

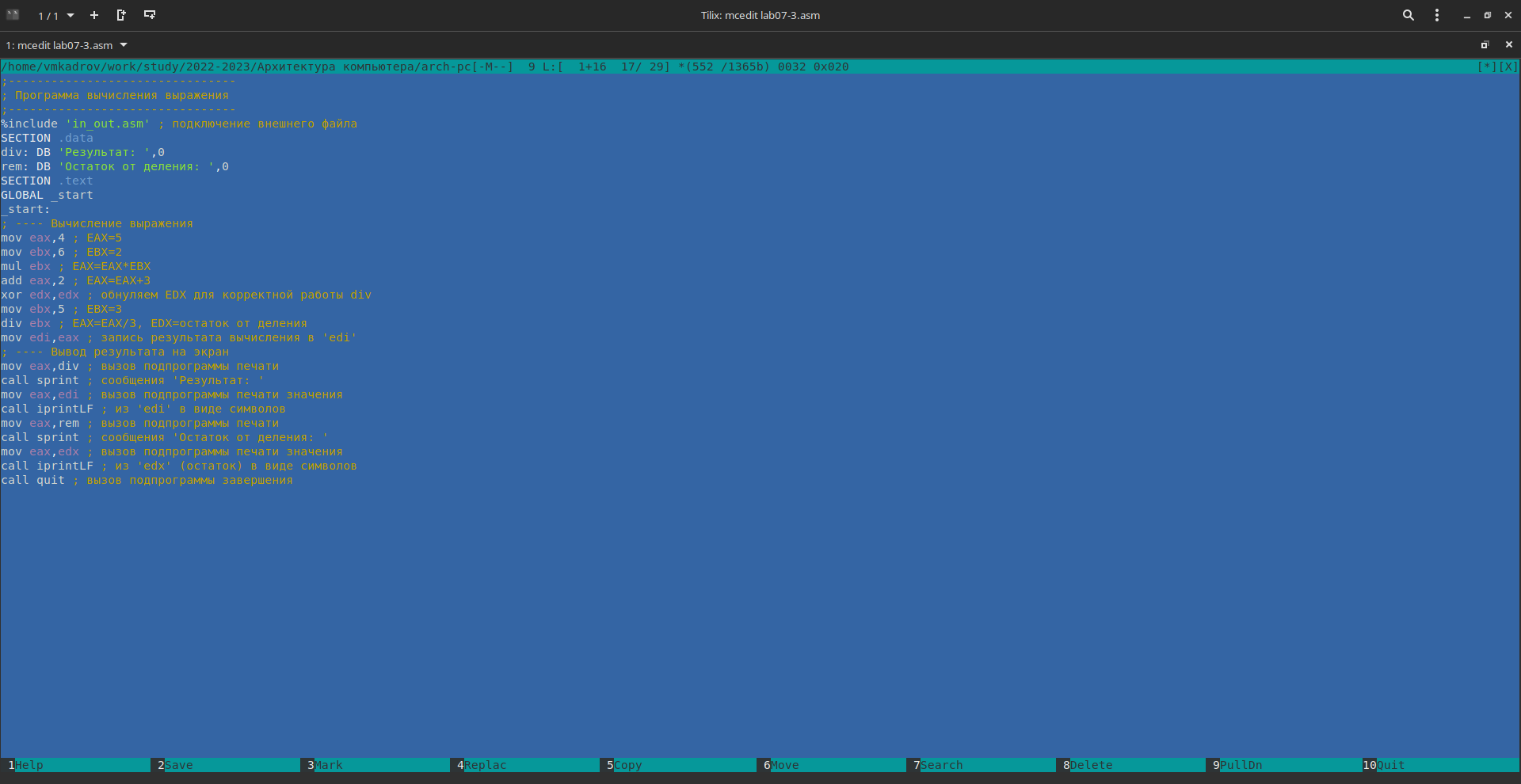


Рис. 12: Измененный код

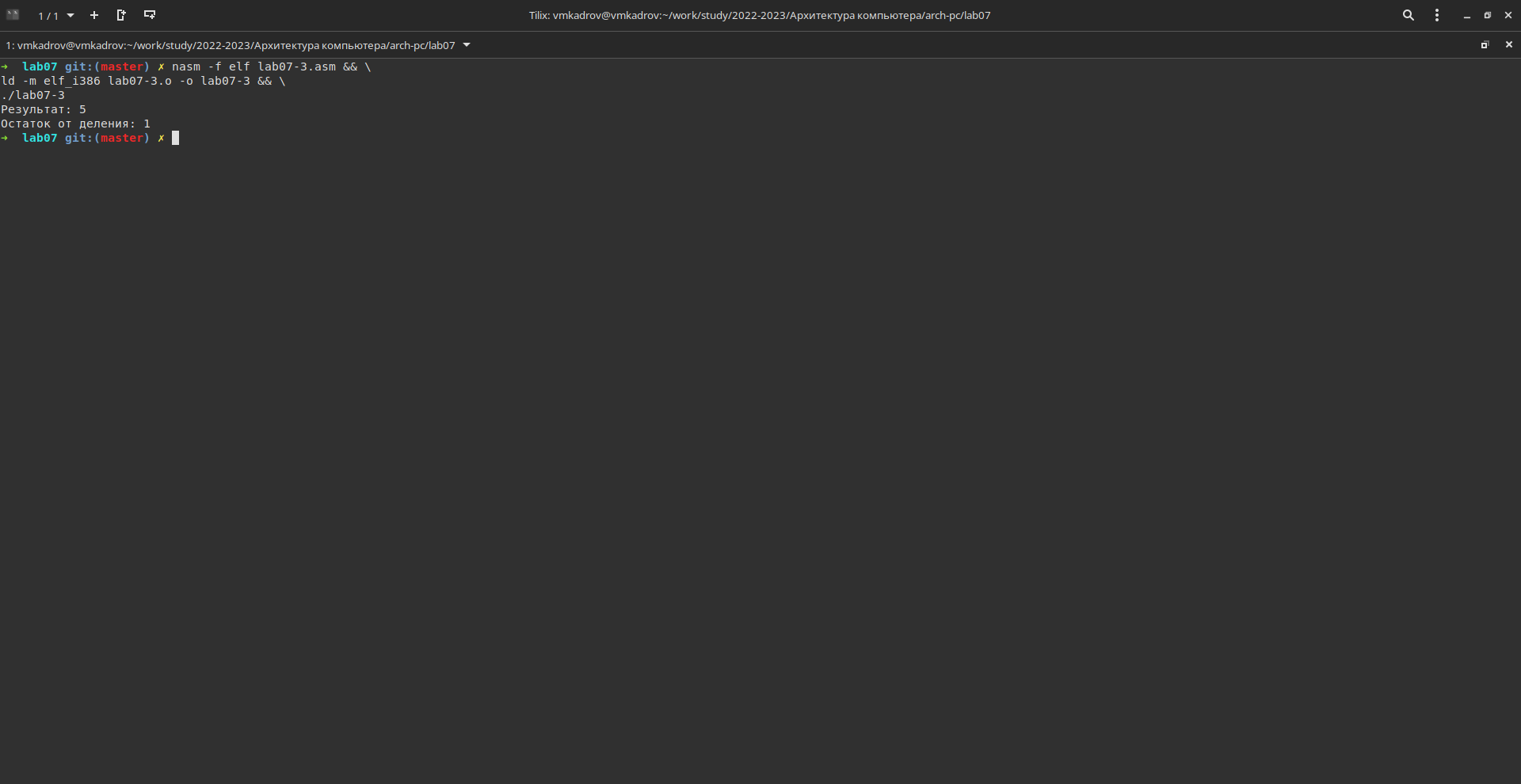


Рис. 13: Выполнение lab07-3

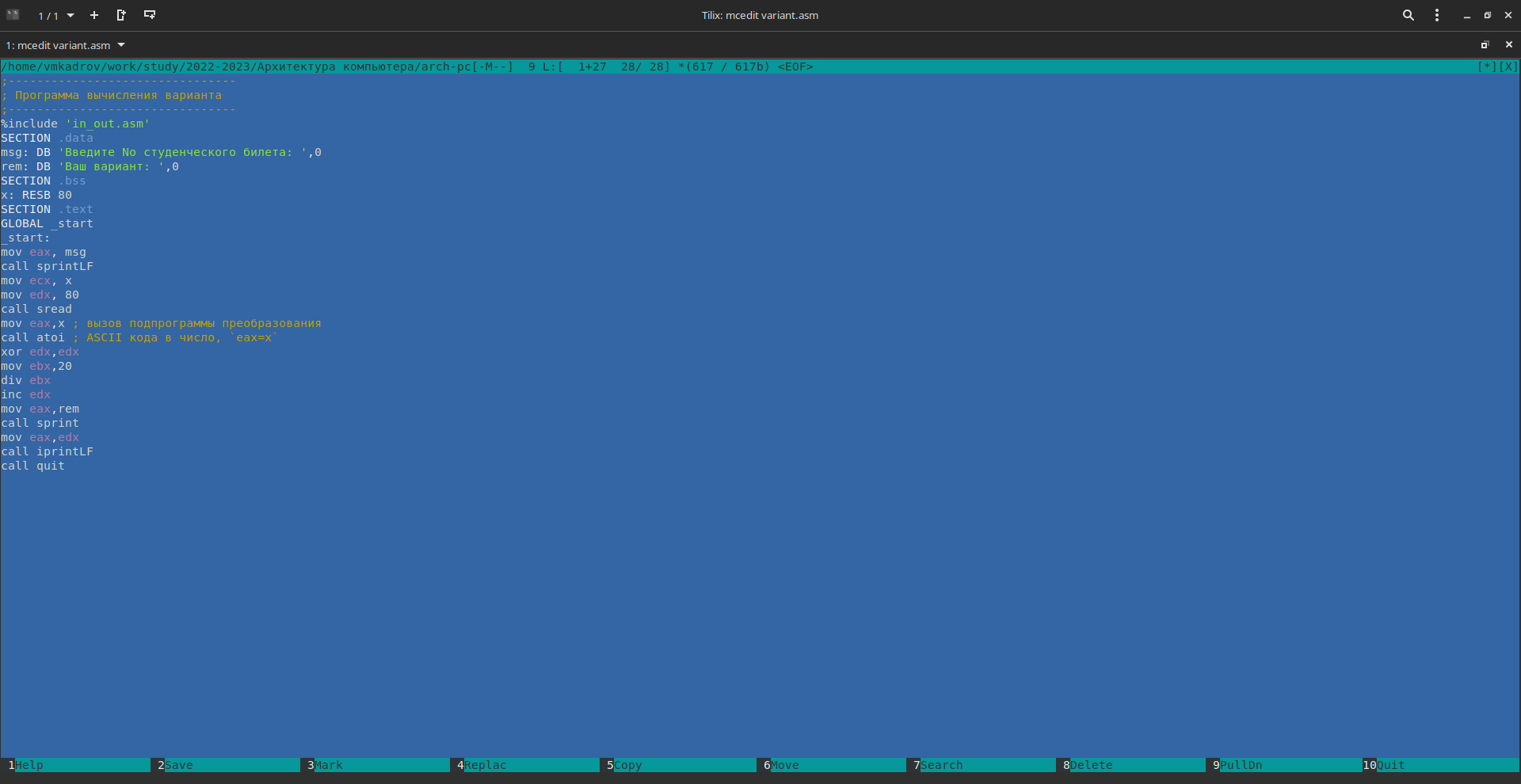


Рис. 14: Код программы

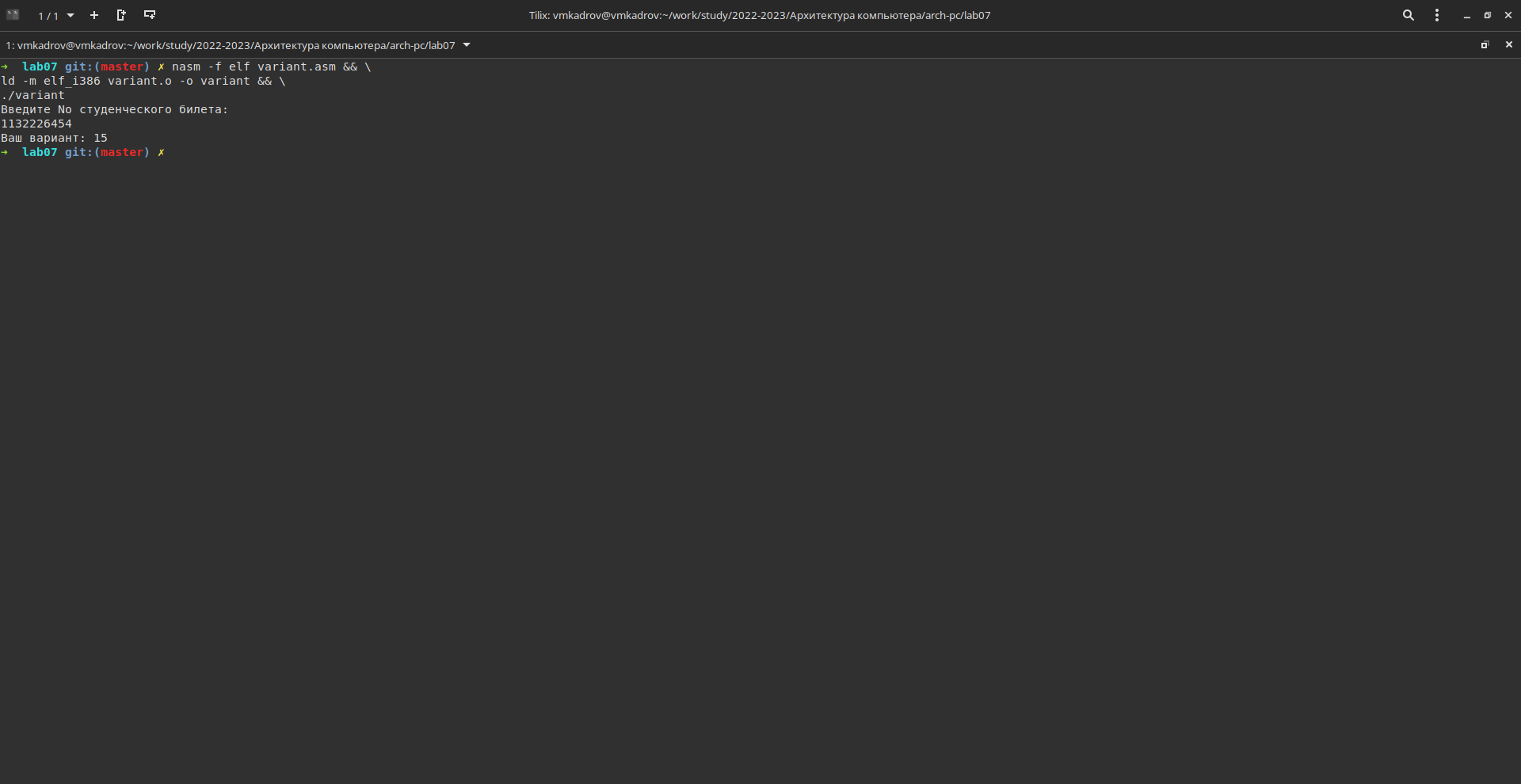


Рис. 15: Номер варианта

# 5 Ответы на вопросы

1. За вывод этого сообщения отвечают строки

mov eax,rem   
call sprint

1. Эти инструкции используются для ввода значения переменной с клавиатуры
2. Функция atoi преобразует значение в регистре из кода в символ ASCII.
3. За вычисление варианта отвечают следующие строки:

xor edx,edx  
mov ebx,20  
div ebx  
inc edx

1. Остаток от деления записывается в регистр edx
2. inc - инкрементация, то есть прибавление единицы к значению в регистре.
3. За вывод результата вычислений отечают строки

mov eax,edx  
call iprintLF

# 6 Задания для самостоятельной работы

Пишем код программы в соответствии с вариантом (рис. 16). Выполняем его и проверяем корректность на двух значениях x (рис. 17).

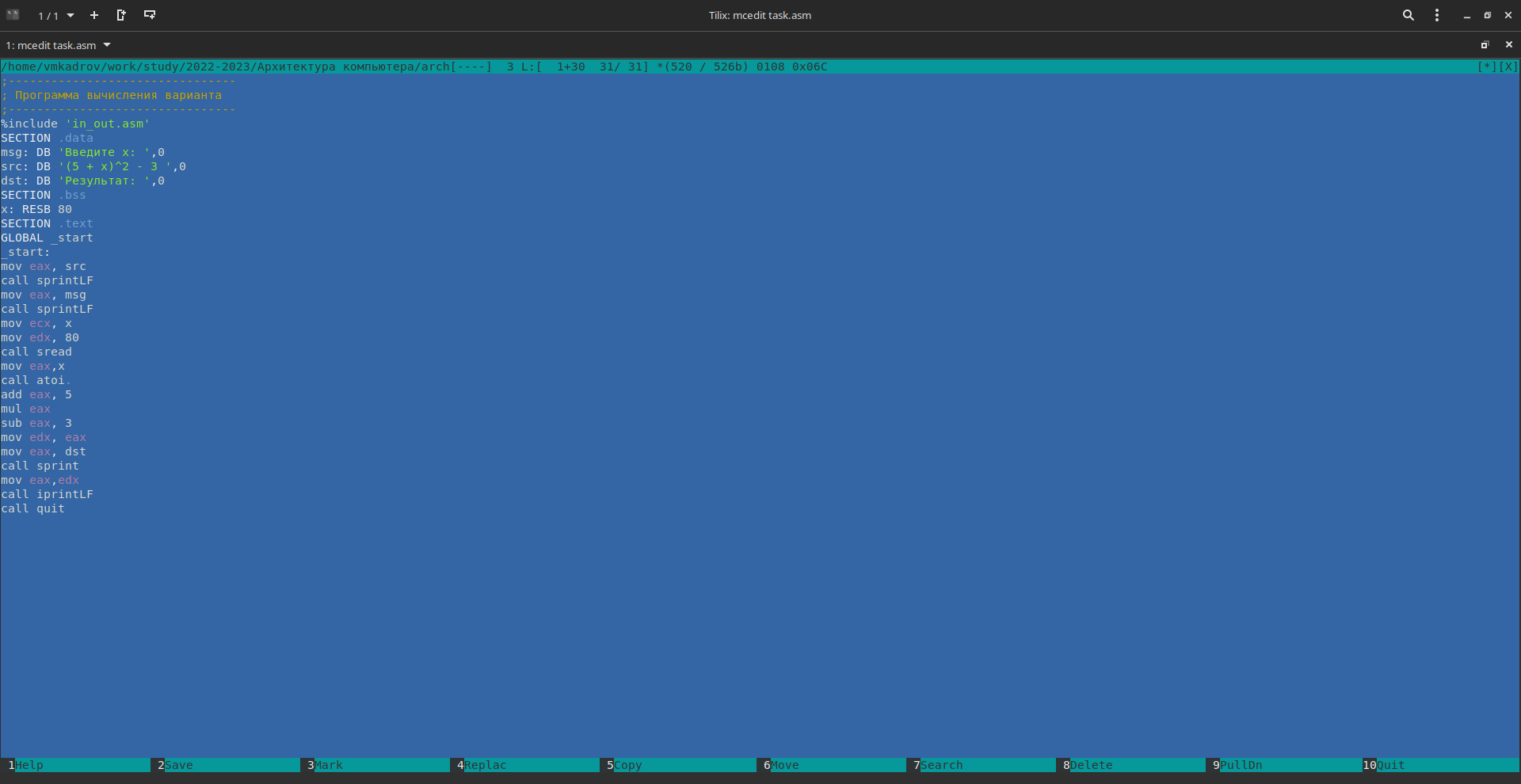


Рис. 16: Код программы

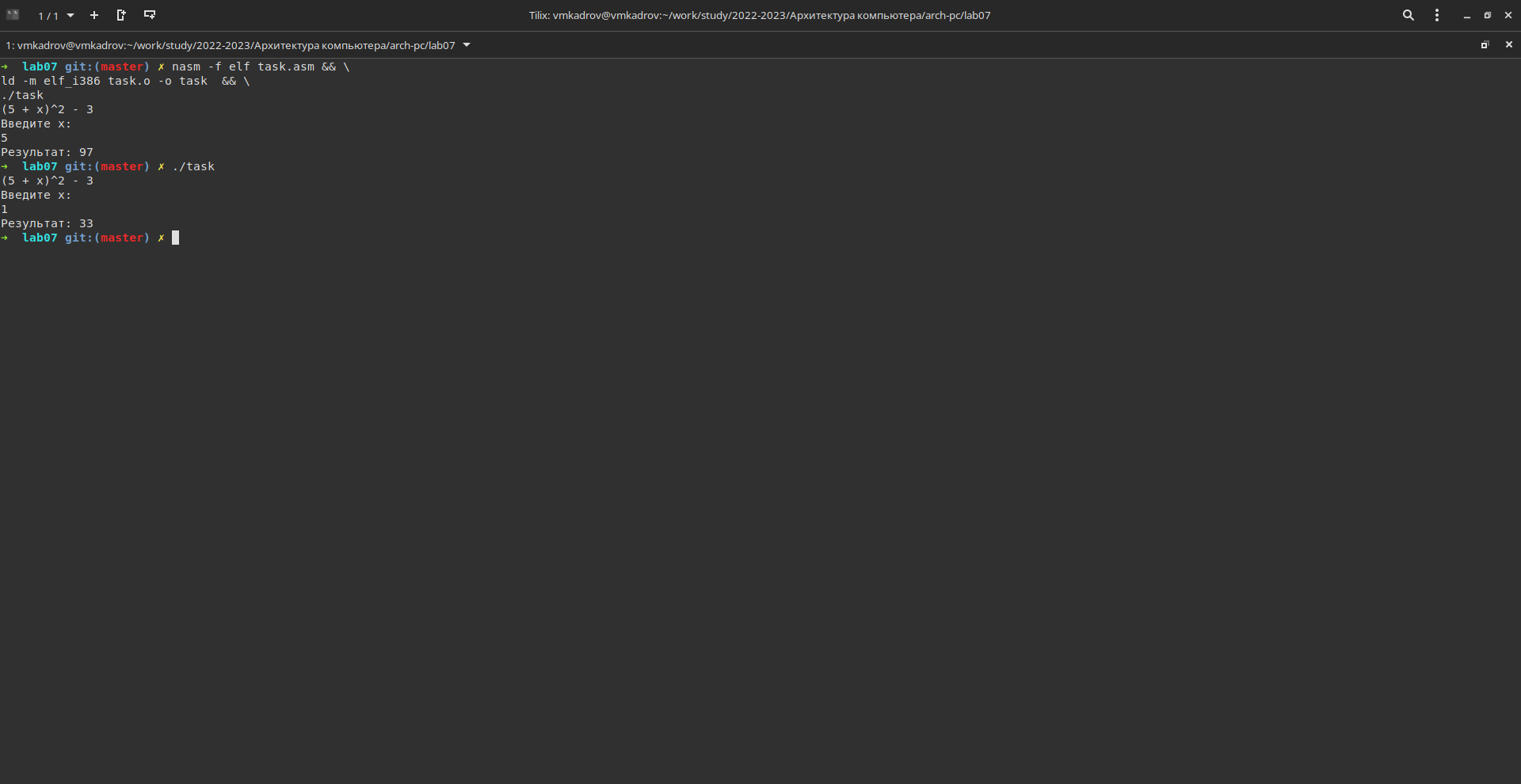


Рис. 17: Результат выполнения

# 7 Выводы

В ходе выполнения лабораторной работы были освоены арифметические инструкции языка ассемблера NASM.