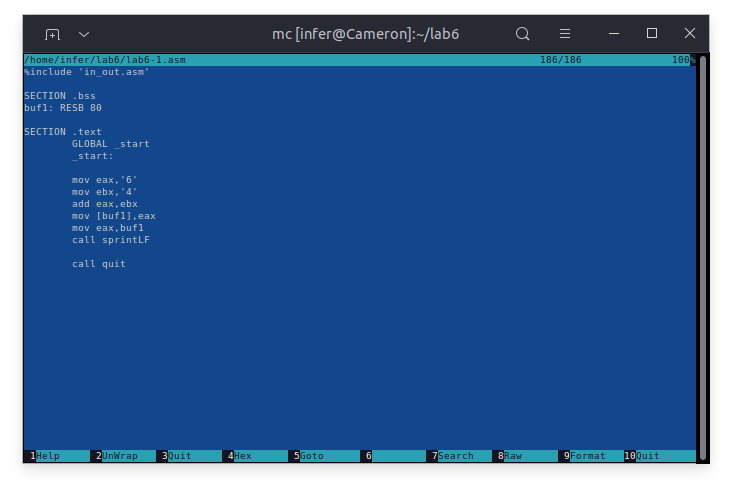
**Лабораторная работа №6. Арифметические операции в NASM**

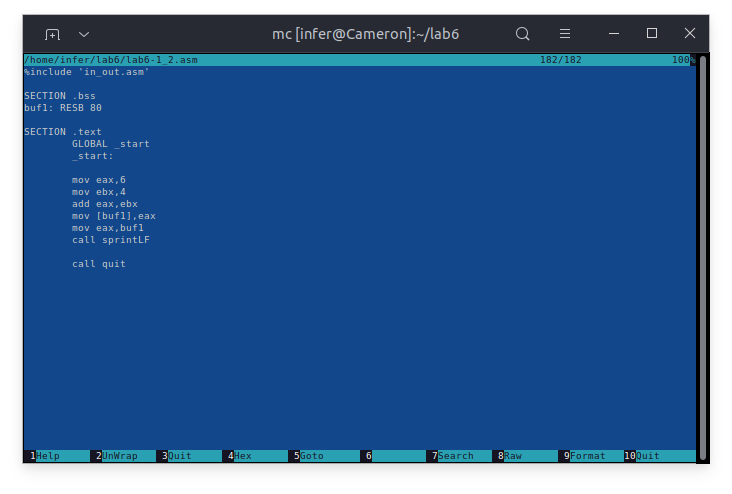
**Цель работы:** Освоение арифметических инструкций языка ассемблера NASM.

**Ход работы.**

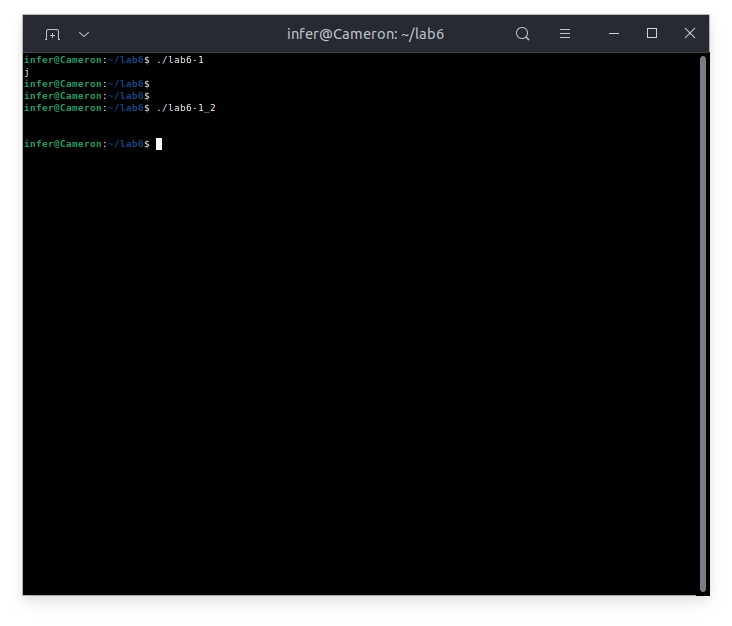
Создадим программу вывода значения регистра eax с исходным кодом, показанным на рисунке 1.

Рисунок 1 — Исходный код программы lab6-1

В ней в регистры eax и ebx записываются символьные константы, которые затем складываются и результат выводится на экран. Также создадим аналогичную программу, в которой идет работа с числовыми константами (рисунок 2)

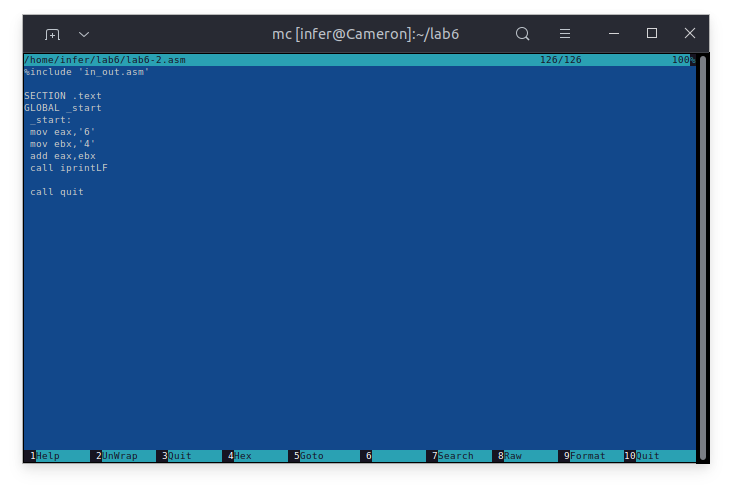
Рисунок 2 — Исходный код программы lab6-1\_2

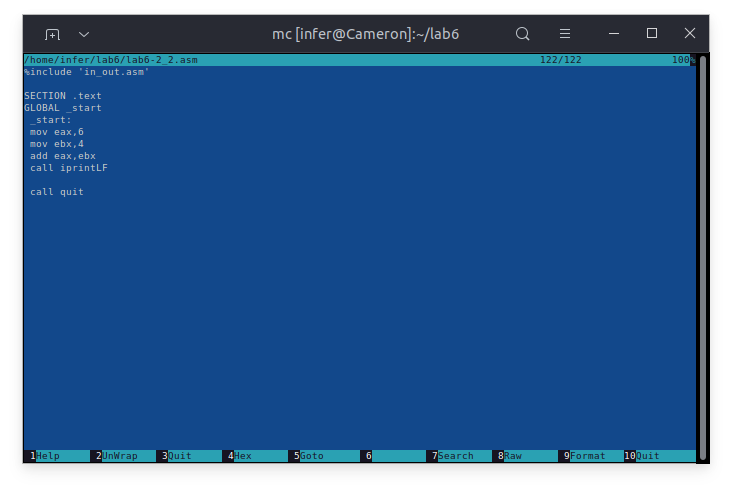
Результат выполнения этих программ показан на рисунке 3.

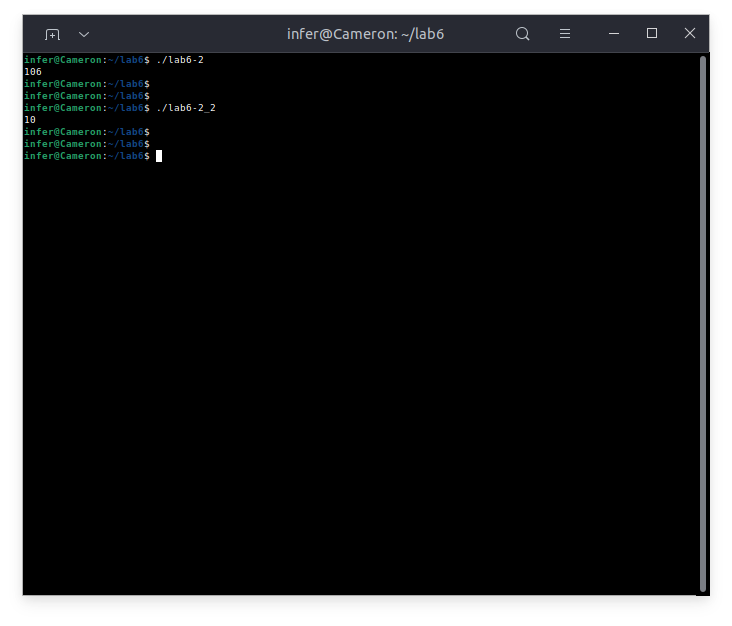
Рисунок 3 — Результат выполнения программ

В первом случае выводится символ j, во втором символ перевода строки. Это связано с тем, сто NASM при сложении символьных констант интерпретирует их как соответвующие числа из таблицы ASCII. Аналогично происходит при выводе на экран (сумма 6 и 4 равна 10, что в таблице ASCII соответствует символу перевода строки).

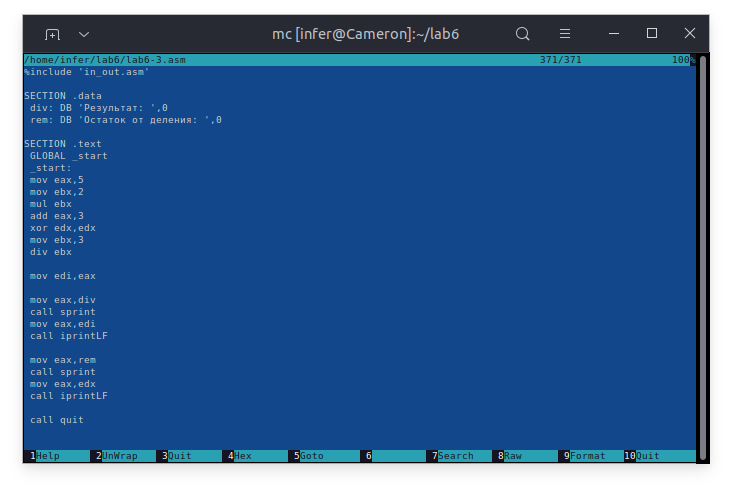
При этом, использование функций iprintLF и iprint из файла in\_out.asm позволяет выводить непосредственно числовые значения. Отличия этих функций заключаются в том, что первая добавляет к выводу символ переноса строки. На рисунках 4 и 5 показаны исходные коды программ, которые демонстрируют работу этих функций. На рисунке 6 показан результат работы программ.

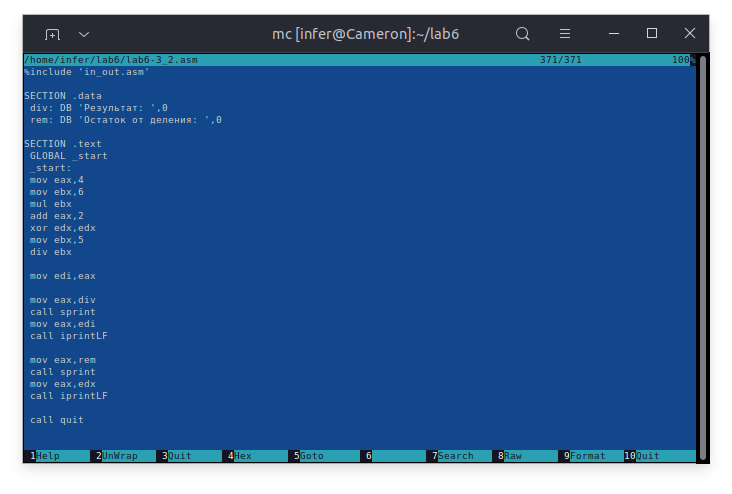
Рисунок 4 — Исходный код программы lab6-2

Рисунок 5 — Исходный код программы lab6-2\_2

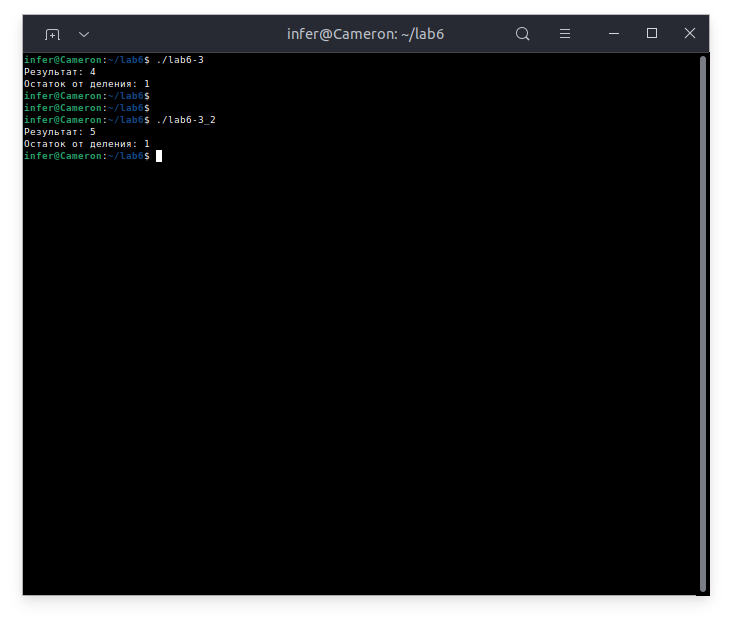
Рисунок 6 – Результат выполнения программ

Создадим программы для вычисления выражений (5\*2+3)/3 и (4\*6+2)/5. Исходные коды программ показаны на рисунках 7 и 8. Как видно из рисунков, код программ по сути идентичен, различия только в записываемых в регистры исходных данных.

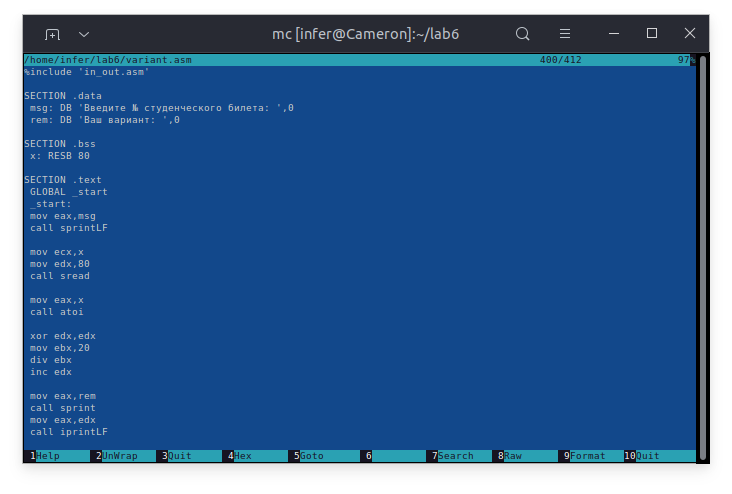
Рисунок 7 — Исходный код программы lab6-3

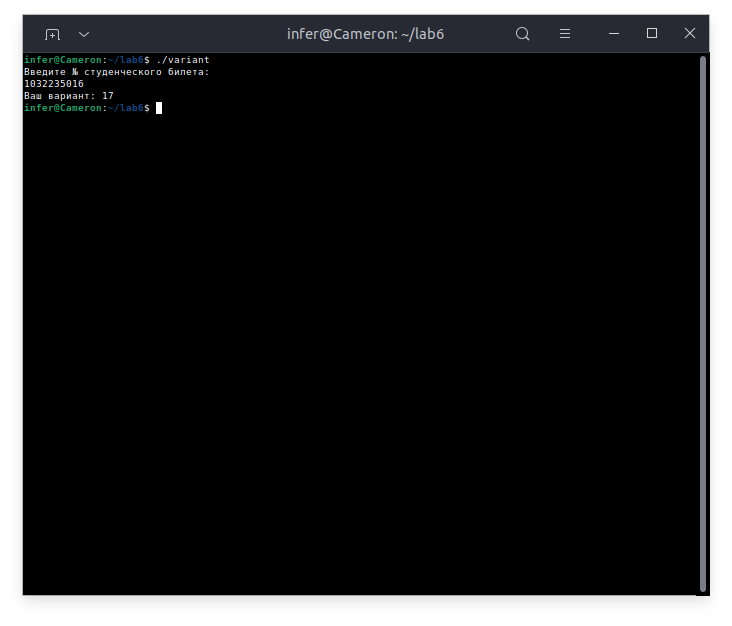
Рисунок 8 — Исходный код программы lab6-3\_2

Результат выполнения программ показан на рисунке 9.

Рисунок 9 — Результат выполнения программ

Далее создадим программу для определения номера варианта по номеру зачетной книжки. Исходный код программы показан на рисунке 10, результат выполнения на рисунке 11.

Рисунок 10 — Исходный код программы variant

Рисунок 11 — Результат выполнения программы variant

Как видно из рисунка 11, моему номеру зачетной книжки соответствует 17 вариант.

Для вывода на экран строки «Ваш вариант» используются команды mov eax,rem и call sprint.

Инструкции mov ecx,x; mov edx,80;call sread используются для считывания символов из консоли в буфер x.

Инструкция call atoi используется для преобразования символов из буфера в числовой вид.

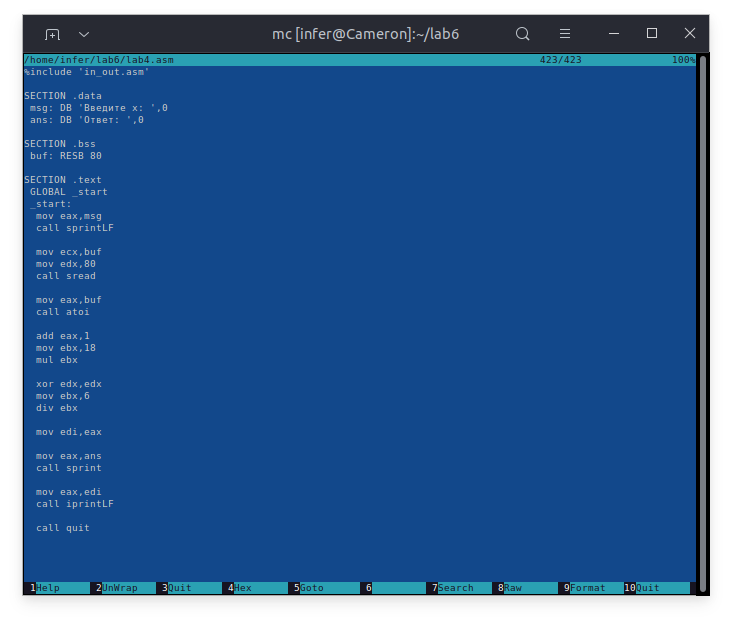
Непосредственно вычисление варианта выполняется инструкциями: mov ebx,20;div ebx;inc edx.

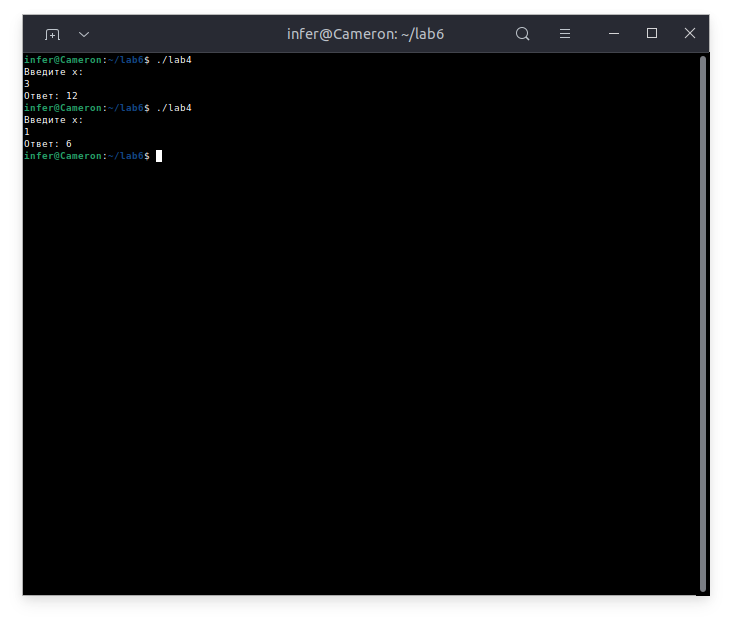
Остаток от деления записывается в регистр edx.

Инструкция inc edx используется для увеличения на единицу значения, находящегося в регистре edx.

За вывод результата вычисления отвечают следующие инструкции: mov eax,edx; call iprintLF..

Далее, согласно полученному номеру варианта создадим программу вычисляющего выражение18\*(x+1)/6 и проверим результат работы взяв x равным 3 и 1. Исходный код программы показан на рисунке 12, результат выполнения на рисунке 13.

Рисунок 12 — Исходный код программы lab4

Рисунок 13 — Результат выполнения программы lab4

Как несложно убедиться, выведенные ответы являются верными.

**Вывод:** В ходе выполнения лабораторной работы я научилась выполнять арифметические операции в NASM.