| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика.**

**ОТЧЕТ**

| **по лабораторной работе №** | 3 |
| --- | --- |

Название: Программирование ветвлений и итерационных циклов

**Дисциплина: Машинно-зависимые языки и основы компиляции**

| Студент |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Я.С. Петрова |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Цель работы:** изучение средств и приемов программирования ветвлений и итерационных циклов на языке ассемблера.

**Задание:** разработать программу на языке ассемблера, вычисляющую целочисленное выражение, представленное на рисунке 1.

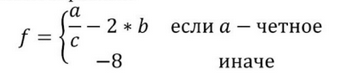


Рисунок 1 — Задание

**Задания 3.2.1-3.2.2**

Прочитайте и проанализируйте свой вариант задания. Разработайте схему алгоритма решения задачи.

Схема алгоритма задачи показана на рисунке 2.

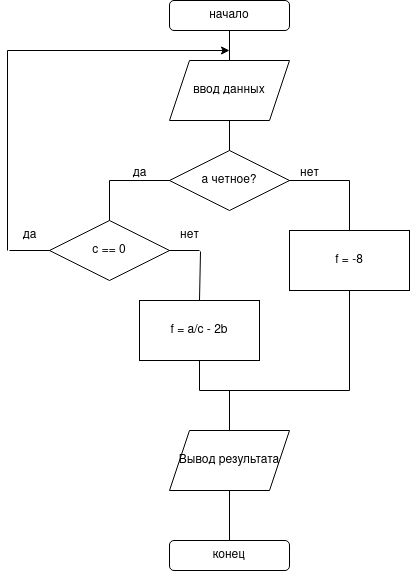


Рисунок 2 — Схема алгоритма решения задачи

**Задания 3.2.3 - 3.2.4**

Напишите соответствующую программу на языке ассемблера. Введите текст программы, выполните ее трансляцию и компоновку.

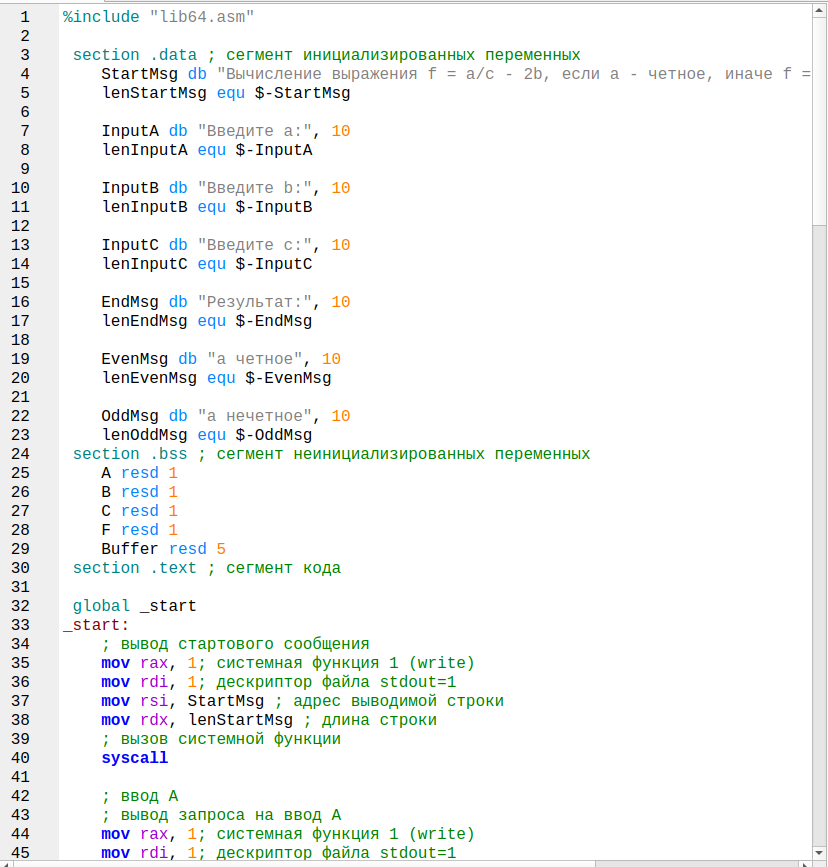


Рисунок 3 –- Код программы (часть 1)

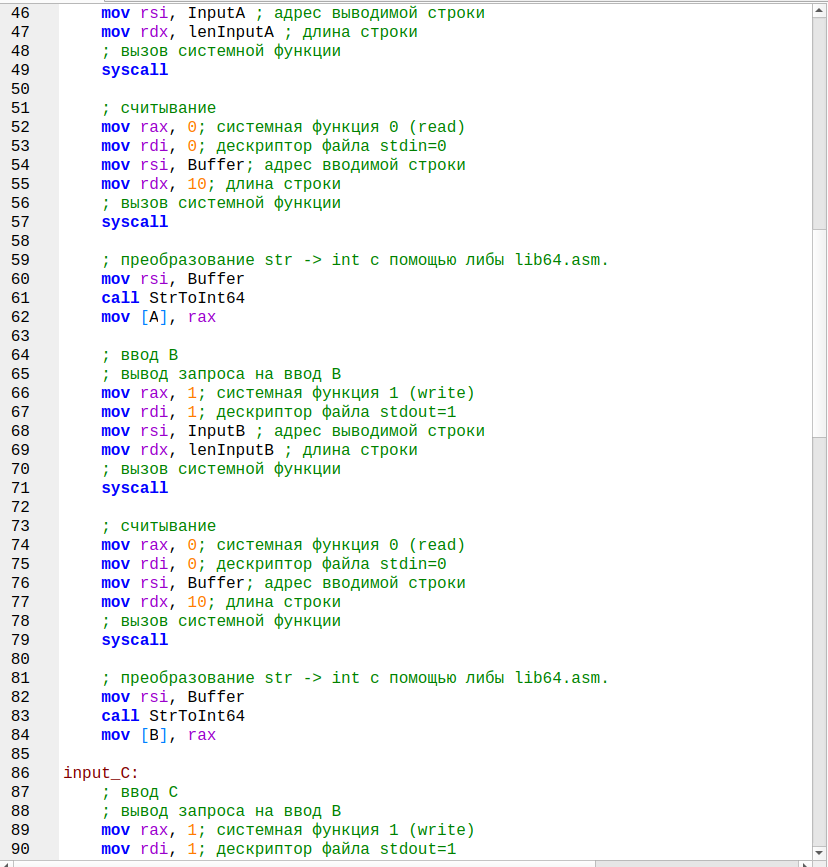


Рисунок 4 — Код программы (часть 2)

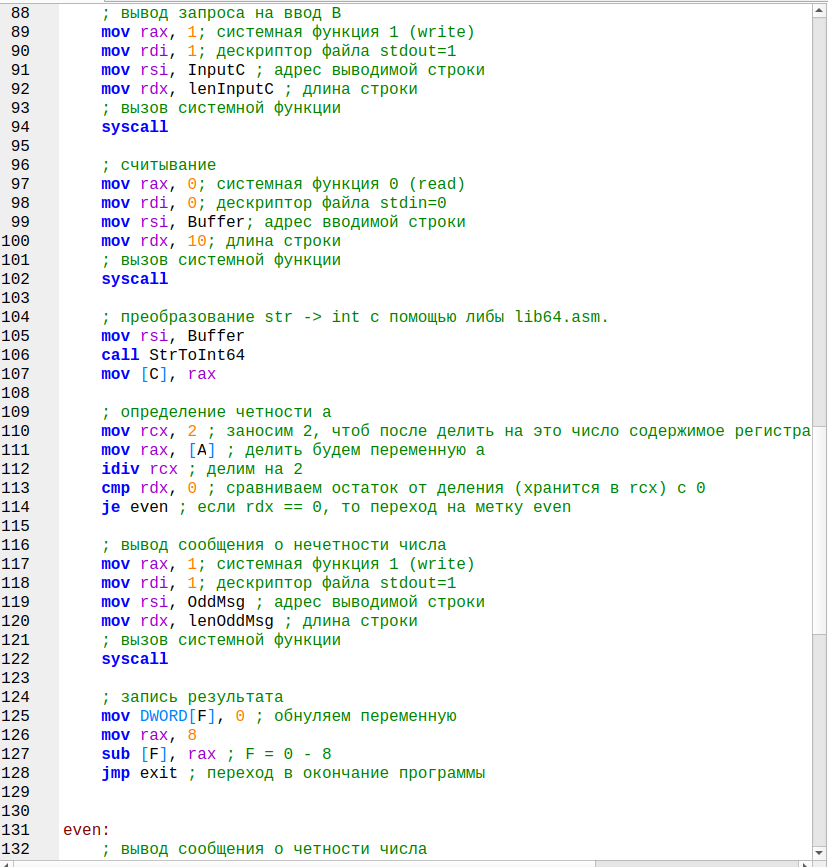


Рисунок 5 — Код программы (часть 3)

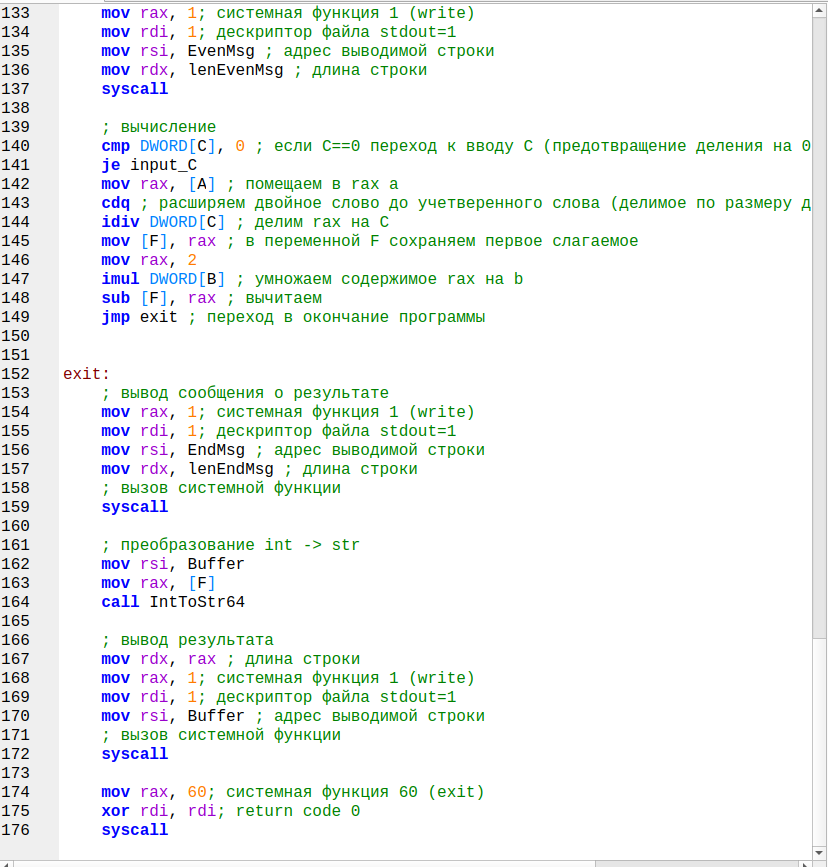


Рисунок 6 — Код программы (часть 4)

**Задания 3.2.5 - 3.2.6** Подберите тестовые данные (не менее 3-х вариантов) и заранее посчитайте на калькуляторе или устно ожидаемый результат. Протестируйте и отладьте программу на выбранных тестовых данных.

Таблица 1. Сравнение ожидаемых и полученных данных

| Исходные данные | Ожидаемый результат | Полученный результат |
| --- | --- | --- |
| a = -3  b = 1  c = 1 | f = -8 |  |
| a = 4  b = 1  c = -1 | f = -6 |  |
| a = 9  b = 2  c = 3 | f = -8 |  |

Отладим программу.

1. **mov rax, [A]**

Операция 1 записывает в регистр eax значение переменной A.



Рисунок 7 — Содержимое регистра rax после выполнения операции 1

1. **idiv DWORD[C]**

Операция 2 делит содержимое регистра rax на операнд и помещает результат вычисления в регистр rax.



Рисунок 8 — Содержимое регистра rax после выполнения операции 2

1. **mov [F], rax**

****

Рисунок 8 — Содержимое переменной после выполнения операции 3

1. **mov rax, 2**

Делит содержимое регистра eax на операнд. Результат помещается в регистр eax.



Рисунок 9 — Содержимое регистра rax после выполнения операции 4

1. **imul DWORD[B]**



Рисунок 10 — Содержимое регистра rax после выполнения операции 5

1. **sub [F], rax**



Рисунок 10 — Содержимое переменной F после выполнения операции 6

**Контрольные вопросы**

**1. Какие машинные команды используют при программировании ветвлений и циклов?**

Ветвления на языке ассемблера программируют с использованием команд условной и безусловной передачи управления. Поскольку команды условного перехода предполагают анализ результата выполнения предыдущих команд, сначала выполняют сравнение – проверку заданного условия. В результате выполнения операции сравнения будут установлены флаги, по которым будет осуществляться переходы.

Командами условной передачи управления являются jle, jge, je, jz и т.д. Командой безусловной передачи управления является команда jmp.

**2. Выделите в своей программе фрагмент, реализующий ветвление. Каково назначение каждой машинной команды фрагмента?**

Фрагмент в программе, реализующий ветвление представлен на рисунке 14.



Рисунок 11 — Ветвление

Для реализации ветвления сперва сравнивается содержимое регистра raх с литералом. При выполнении команды сравнения, устанавливаются или сбрасываются флаги, указывающие на результат сравнения. Далее, основываясь на установившихся флагах, происходит условный переход по меткам. Если число в регистре rах оказалось равным нулю, то произойдет переход по метке even.

**3. Чем вызвана необходимость использования команд безусловной передачи управления?**

Необходимость безусловной передачи управления объясняется потребностью выполнять разные блоки программы при разных исходных данных.

**4. Поясните последовательность команд, выполняющих операции ввода-вывода в вашей программе. Чем вызвана сложность преобразований данных при выполнении операций ввода-вывода?**

Сперва осуществляется вывод сообщения в терминал в виде, имеющее следующее содержание “Введите Х”. Затем пользователь вводит число, однако, программа воспринимает это число как строковое значение. Для преобразования строки в число используется функция StrToInt. После этого число записывается в переменную и может использоваться для операций над целочисленными значениями. При завершении работы с переменным в виде числа и для вывода в терминал, необходимо снова преобразовать число. В этот раз из целочисленного значения в строковое посредством функции IntToStr.

**Вывод:** в результате выполнения лабораторной работы были освоены основные принципы ветвлений, написана программа вычисления выражения в зависимости от введенных данных.