*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«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)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № 3**

**Вариант 22**

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

**Название лабораторной работы:** **Программирование ветвлений и циклов**

Студент гр. ИУ6-44  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шумаков А.А.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Данилюк С.С.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

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

**Задание**

Разработать программу на языке ассемблера, которая вычисляет заданное выражение.

Исходное выражение:

**Выполнение**

В первую очередь приведем на рисунке 1 схему алгоритма.

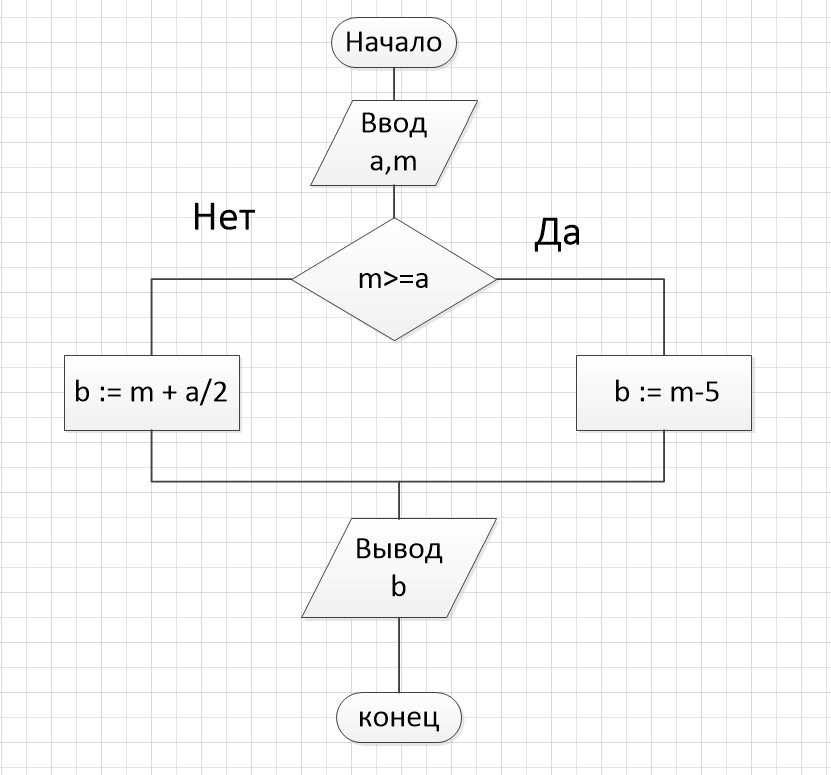


Рисунок : Схема алгоритма

Приведем код данной программы с комментариями:

*; Template for console application*

*.586*

*.MODEL flat, stdcall*

*OPTION CASEMAP:NONE*

*Include kernel32.inc*

*Include masm32.inc*

*IncludeLib kernel32.lib*

*IncludeLib masm32.lib*

*.CONST*

*MsgExit DB 13,10,"Press Enter to Exit",0AH,0DH,0*

*.DATA*

*Zapros DB 13,10,'Input a',13,10,0*

*Zapros2 DB 13,10,'Input m',13,10,0*

*ResTrue DB 13,10,'Find:m-5',13,10,0*

*ResFalse DB 131,10,'Find:m+a/2',13,10,0*

*Result DB 'b = '*

*ResStr DB 16 DUP (' '),0*

*.DATA?*

*Buffer DB 10 DUP (?)*

*inbuf DB 100 DUP (?)*

*a sWord ?*

*m sword ?*

*b sword ?*

*.CODE*

*Start:*

*Invoke StdOut,ADDR Zapros*

*Invoke StdIn,ADDR Buffer,LengthOf Buffer*

*Invoke StripLF,ADDR Buffer*

*Invoke atol,ADDR Buffer*

*mov dword PTR a,EAX*

*Invoke StdOut,ADDR Zapros2*

*Invoke StdIn,ADDR Buffer,LengthOf Buffer*

*Invoke StripLF,ADDR Buffer*

*Invoke atol,ADDR Buffer*

*mov dword PTR m,EAX*

*;вычислить m-5 или m+a/2*

*;Начало вычислений*

*;AX:=m*

*cmp AX,a ;Начало ветвления*

*jge metka*

*mov CX,AX; CX:=m*

*XOR EAX,EAX*

*mov AX,a ; AX:=a*

*mov BX,2*

*cwd*

*idiv sword PTR BX ; AX:=a/2*

*add CX,AX; CX:=m+a/2*

*mov b,CX*

*Invoke StdOut,ADDR ResFalse*

*jmp konec*

*metka: sub AX,5 ;AX:=AX-5*

*mov b,AX ;b:=m-5*

*Invoke StdOut,ADDR ResTrue*

*konec: Invoke dwtoa,b,ADDR ResStr*

*Invoke StdOut,ADDR Result*

*XOR EAX,EAX*

*Invoke StdOut,ADDR MsgExit*

*Invoke StdIn,ADDR inbuf,LengthOf inbuf*

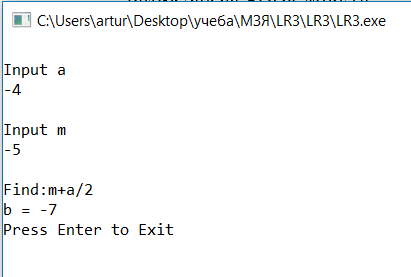
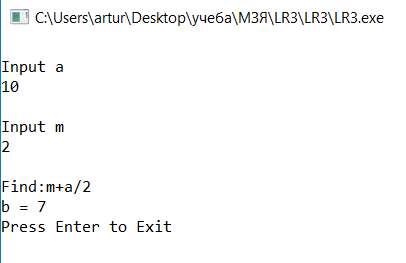
*Invoke ExitProcess,0*

*End Start*

На рисунке 2 приведены результаты выполнения программы

**Тесты программы**

|  |  |  |  |
| --- | --- | --- | --- |
| Исходные данные | | Ожидаемый результат | Полученный результат |
| m | a |
| -5 | -4 | -7 | -7 |
| 2 | 10 | 7 | 7 |
| -4 | -4 | -9 | -9 |
| 10 | 2 | 5 | 5 |

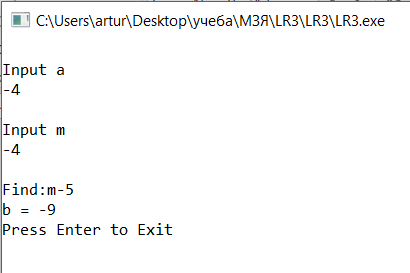
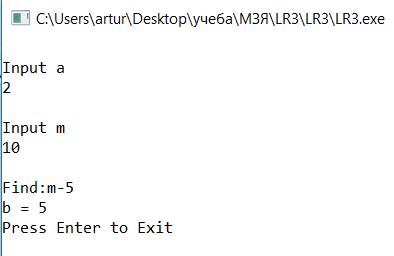
 

Рисунок : Результаты выполнения программы

**Вывод**

Таким образом, разработана программа, вычисляющая заданное выражение в зависимости от значения входных параметров, то есть была реализована конструкция ветвления.