*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**

**Дисциплина:** Машинно-зависимые языки программирования

**Название лабораторной работы:**

Программирование ветвлений и циклов

Студент гр. ИУ6-42  **\_\_\_\_\_\_03.04.2018 Бурлаков Александр Сергеевич**

(Подпись, дата) (И.О. Фамилия)

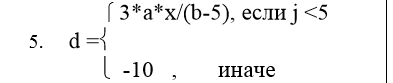
Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2018

**ЗАДАНИЕ**

Разработать программу на языке ассемблера, которая вычисляет заданное выражение.



**ОСНОВНАЯ ЧАСТЬ**

1. **Схема алгоритма**

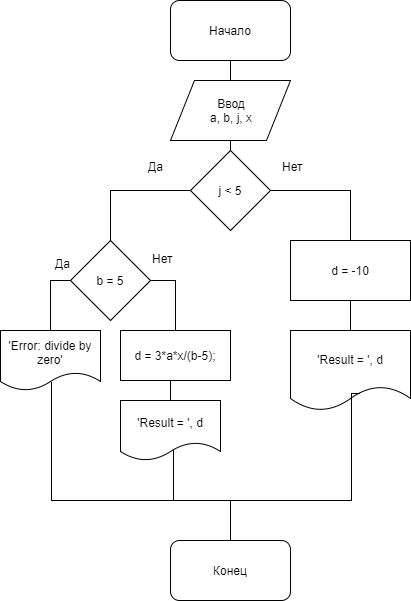


Рисунок 1 – схема алгоритма разрабатываемой программы

1. **Код разработанной программы**

; 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

A SWORD 0

B SWORD 0

J SWORD 0

X SWORD 0

Msg1 DB 'Input a:', 13, 10, 0

Msg2 DB 'Input b:', 13, 10, 0

Msg3 DB 'Input j:', 13, 10, 0

Msg4 DB 'Input x:', 13, 10, 0

buffer DB 10 dup ('0')

MsgOut DB 'Result = '

ResStr DB 8 dup (' ')

MsgErr DB 'Error: divide by zero', 13, 10, 0

.DATA?

inbuf DB 100 DUP (?)

D SWORD ?

.CODE

Start:

; a = 3\*a\*x/(b-5, b > 0

; a = 3 , b <= 0

Invoke StdOut, ADDR Msg1

Invoke StdIn, ADDR buffer, LengthOf buffer

Invoke StripLF, ADDR buffer

Invoke atol, ADDR buffer

mov A, AX

Invoke StdOut, ADDR Msg2

Invoke StdIn, ADDR buffer, LengthOf buffer

Invoke StripLF, ADDR buffer

Invoke atol, ADDR buffer

mov B, AX

Invoke StdOut, ADDR Msg3

Invoke StdIn, ADDR buffer, LengthOf buffer

Invoke StripLF, ADDR buffer

Invoke atol, ADDR buffer

mov J, AX

Invoke StdOut, ADDR Msg4

Invoke StdIn, ADDR buffer, LengthOf buffer

Invoke StripLF, ADDR buffer

Invoke atol, ADDR buffer

mov X, AX

cmp J, 5

jl less ; if (b <= 0) {

mov D, -10

Invoke dwtoa, D, ADDR ResStr

Invoke StdOut, ADDR MsgOut

jmp continue

less:

cmp B, 5

je equal

mov AX, B

sub AX, 5

mov B, AX

mov AX, A

mov CX, 3

imul CX

imul X

idiv B

mov D, AX

Invoke dwtoa, D, ADDR ResStr

Invoke StdOut, ADDR MsgOut

jmp continue

equal:

Invoke StdOut, ADDR MsgErr

continue:

Invoke StdOut,ADDR MsgExit

Invoke StdIn,ADDR inbuf,LengthOf inbuf

Invoke ExitProcess,0

End Start

1. **Тестирование программы**

Таблица 1 – результаты тестирования программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | J | X | Ожидаемые результаты | Полученные результаты |
| 5 | 6 | 8 | 7 | -10 | -10 |
| 2 | 5 | 6 | 5 | -10 | -10 |
| 7 | 5 | 4 | 0 | Error: divide by zero | Error: divide by zero |
| 3 | 4 | 3 | 5 | -45 | -45 |
| 6 | 12 | 3 | 6 | 15 | 15 |

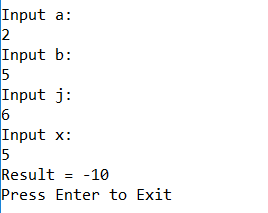
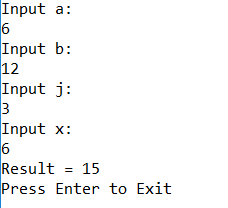
 

Рисунок 2 – Результаты тестирования программы

**ВЫВОД**

1. Разработан алгоритм, позволяющий решить данное задание.
2. Написан код программы по разработанному алгоритму.
3. Произведено тестирование программы, тестирование показало корректность работы программы.