|  |  |
| --- | --- |
| 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\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**по лабораторной работе № 3**

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

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

Студент гр. ИУ6-41Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Самодурова**

(Подпись, дата) (И.О. Фамилия)

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

(Подпись, дата) (И.О. Фамилия)

Москва, 2023

**Цель:** изучение средств и приемов программирования ветвлений

и итерационных циклов на языке ассемблера.

**Текст задания:**

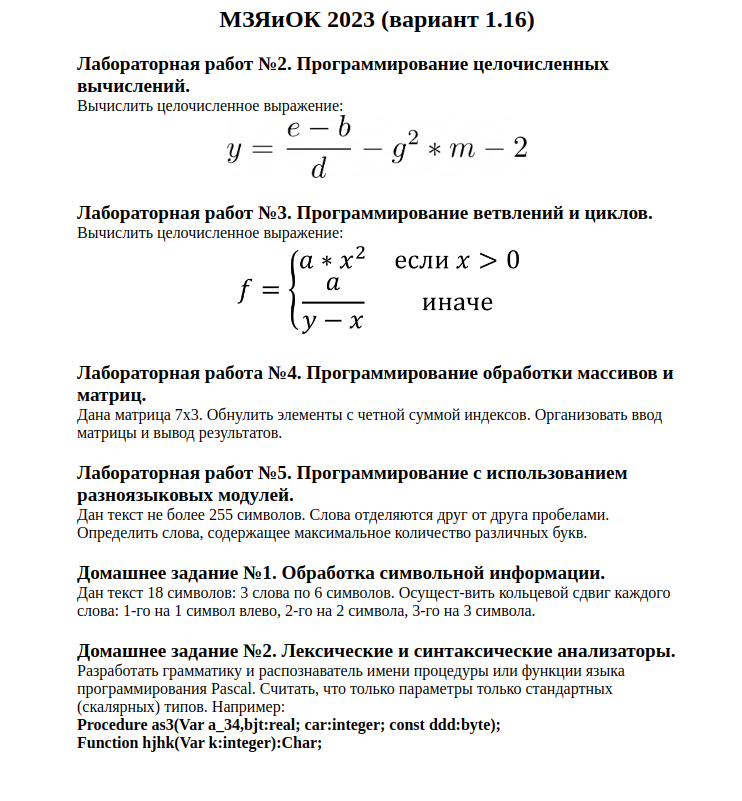
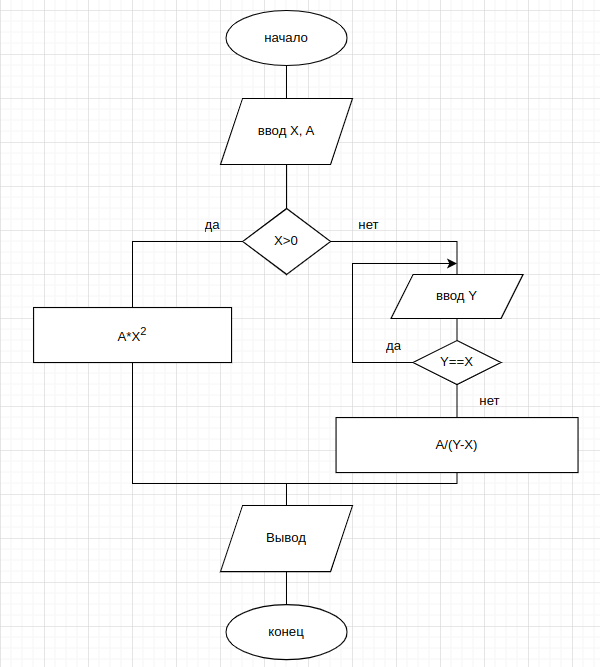


Схема программы на рисунке 1

Рисунок 1 — схема программы

**Код программы:**

%include "io64.asm"

section .data

ExitMsg db "Press Enter to Exit",10

lenExit equ $-ExitMsg

InputMsg db "Input X",10

lenInput equ $-InputMsg

AMsg db "Inter A",10

lenA equ $-AMsg

YMsg db "Inter Y",10

lenY equ $-YMsg

NMsg db "Inter N",10

lenN equ $-NMsg

; A dd 90

; Y dd 3

N dd 0

; X dd -8

section .bss

InBuf resb 10

lenIn equ $-InBuf

X resd 1

A resw 1

Y resw 1

section .text

global \_start

\_start:

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, InputMsg ; адрес выводимой строки

mov rdx, lenInput ; длина строки

syscall ; вызов системной функции

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [X], rax

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, AMsg ; адрес выводимой строки

mov rdx, lenA ; длина строки

syscall ; вызов системной функции

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [A], rax

mov eax, [X]

cmp eax, [N] ; сравнение A и B

jle LESS ; если первое меньше или равно

mov eax, [X]

imul eax ;q\*q

mov ebx, [A]

imul ebx

jmp short CONTINUE ; безусловный переход

LESS:

input: mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, YMsg ; адрес выводимой строки

mov rdx, lenY ; длина строки

syscall ; вызов системной функции

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

call StrToInt64

mov [Y], rax

mov ax, [Y]

mov bx, [X]

sub ax, bx

mov bx, 0

cmp ax, bx

je input

mov ebx, [Y]

sub ebx, [X]

mov eax, [A]

cwd

idiv ebx

CONTINUE:

cwde

mov rsi, InBuf

call IntToStr64

; write

mov rdx, rax

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

syscall

mov rax, 1 ; системная функция 1 (write)

mov rdi, 1 ; дескриптор файла stdout=1

mov rsi, ExitMsg ; адрес выводимой строки

mov rdx, lenExit ; длина строки

syscall ; вызов системной функции

; read

mov rax, 0 ; системная функция 0 (read)

mov rdi, 0 ; дескриптор файла stdin=0

mov rsi, InBuf ; адрес вводимой строки

mov rdx, lenIn ; длина строки

syscall ; вызов системной функции

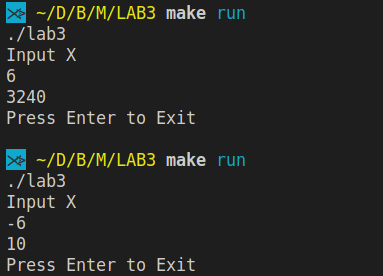
; exit

mov rax, 60 ; системная функция 60 (exit)

xor rdi, rdi ; return code 0

syscall ; вызов системной функции

Результат работы программы на рисунке 2

Рисунок 2 — результат выполнения

**Контрольные вопросы**

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

Для организации ветвлений и циклов -пока и -до используются команды безусловного перехода: jmp, и условного перехода: jz, jnz, je, jne, jl, jg, ja, jb и т. д.

Для программирования счетных циклов также используется команда loop, loope, loopne.

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

...

cmp eax, [N] ; сравнение A и B

jle LESS ; если первое меньше или равно

mov eax, [X]

imul eax ;q\*q

mov ebx, [A]

imul ebx

jmp short CONTINUE ; безусловный переход

LESS:

mov ebx, [Y]

sub ebx, [X]

mov eax, [A]

cwd

idiv ebx

CONTINUE: ...

3. Чем вызвана необходимость использования команд безусловной передачи управления?

Необходимость использования команд безусловной передачи управления вызвана тем, что при попадании в одну из веток ветвления, не должно быть передачи управления в другую ветку. Это невозможно без команд безусловной передачи управления, так иначе выполнение идет последовательно.

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

При вводе значений в программу сначала мы записываем данные в буфер в символьном формате, а затем вызываем подпрограмму перевода данных в целочисленный формат.

При выводе действия симметричны, то есть вызываем подпрограмму перевода данных в символьный формат.

Это связано с тем, что системные функции ввода-вывода могут работать только с символьными данными.

**Вывод:** в ходе выполнения данной лабораторной работы я изучила форматы машинных команд, команд ветвлений и циклов и программирование ветвлений и циклов.