|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №7**

**«Команды передачи управления. Циклы»**

**ДИСЦИПЛИНА: «Машинно-зависимые языки программирования»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-32Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Карельский М.К. )  (Подпись) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Амеличева К.А. )  (Подпись) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2021

**Цель:** приобретение навыков написания программ с циклами на языке Ассемблер.

**Задачи:**

Создать программу обработки числовых массивов используя циклические структуры и макросы для ввода и вывода десятичных чисел.

**Вариант 2**

1. Организовать ввод массива 8, 0, -45, 98, -15, 2, 19, -72, 35, -4 с клавиатуры.
2. В заданном числовом массиве определить среднее арифметическое отрицательных элементов.
3. В заданном числовом массиве удвоить элементы с нечетными индексами.
4. В заданном числовом массиве переставить местами соседние элементы с четными и нечетными индексами.

**Блок-схемы:**



**Рисунок 1.** Пункт 1



**Рисунок 2.** Пункт 2 **Рисунок 3.** Пункт 3



**Рисунок 4.** Пункт 4

**Листинг:**

.model small

.386

.stack 100h

.data

array dw 10 dup(0)

buffer db 6 dup(0)

negNum db 0

negSum dw 0

arrayInput db 'Input 8, 0, -45, 98, -15, 2, 19, -72, 35, -4', 10, 13, '$'

numberInput db '>>> ', '$'

arrayOutput db 'Result: ', '$'

averageOutput db 'Average: ', '$'

mInput macro buffer, size

LOCAL NEGATIVE

LOCAL POSITIVE

LOCAL CALCULATION

LOCAL POWER

LOCAL CYCLE

LOCAL END\_CALCULATION

LOCAL END\_INPUT

mov buffer, size

inc buffer

mov ah, 0Ah

lea dx, buffer

int 21h

sub ax, ax

cmp buffer+1, 0

je END\_INPUT

mov ax, 1

sub dx, dx

mov dl, buffer+1

dec dx

mov bx, 10

cmp buffer+2, '-'

je NEGATIVE

jmp POSITIVE

NEGATIVE:

push -1

mov si, 3

dec dx

jmp POWER

POSITIVE:

push 1

mov si, 2

POWER:

cmp dx, 0

jle CALCULATION

mul bl

dec dx

jmp POWER

CALCULATION:

mov cx, ax

sub bx, bx

CYCLE:

sub ax, ax

mov al, buffer[si]

sub al, 30h

mul cx

add bx, ax

mov ax, cx

mov cx, 10

div cx

mov cx, ax

inc si

mov dx, si

sub dl, buffer+1

cmp dl, 2

je END\_CALCULATION

jmp CYCLE

END\_CALCULATION:

pop dx

mov ax, bx

imul dx

END\_INPUT:

endm

mPrintString macro string

mov ah, 09h

mov dx, offset string

int 21h

endm

mPrint macro symbol

mov ah, 9

mov al, symbol

mov bl, 0Fh

mov cx, 1

int 10h

endm

mIncCursor macro

mov ah, 3

mov bh, 0

int 10h

inc dl

mov ah, 2

int 10h

endm

mNextLine macro

mov ah, 3

mov bh, 0

int 10h

inc dh

mov dl, 0

mov ah, 2

int 10h

endm

mOutput macro

LOCAL NEGATIVE

LOCAL SPLIT

LOCAL OUTPUT

mov si, 10

sub di, di

cmp ax, 0

jl NEGATIVE

jmp SPLIT

NEGATIVE:

neg ax

push ax

mPrint '-'

mIncCursor

pop ax

SPLIT:

inc di

sub dx, dx

div si

add dx, 30h

push dx

cmp ax, 0

jg SPLIT

OUTPUT:

pop ax

mPrint al

mIncCursor

dec di

cmp di, 0

jg OUTPUT

endm

.code

Start:

mov ax, @data

mov ds, ax

sub ax, ax

mov ax, 600h

mov bh, 7

mov cx, 0

mov dx, 184Fh

int 10h

mov ah, 2

mov dh, 0

mov dl, 0

mov bh, 0

int 10h

mPrintString arrayInput

sub di, di

ARRAY\_INPUT:

mPrintString numberInput

mInput buffer, 3

cmp al, 0

jge POSITIVE

add negSum, ax

inc negNum

POSITIVE:

mov cx, ax

mov ax, di

mov bx, 4

div bl

cmp ah, 0

mov ax, cx

je EVEN\_NUMBER

mov bx, 2

imul bl

mov si, di

dec si

dec si

mov bx, array[si]

mov array[si], ax

mov array[di], bx

jmp NEW\_LINE

EVEN\_NUMBER:

mov array[di], ax

NEW\_LINE:

inc di

inc di

mNextLine

cmp di, 20

jne ARRAY\_INPUT

sub di, di

mPrintString arrayOutput

ARRAY\_OUTPUT:

sub ax, ax

mov ax, array[di]

push di

mOutput

mIncCursor

pop di

inc di

inc di

cmp di, 20

jne ARRAY\_OUTPUT

mNextLine

mPrintString averageOutput

mov ax, negSum

mov bl, negNum

idiv bl

cbw

mOutput

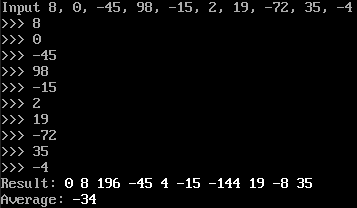
mov ax, 4C00h

int 21h

end start

END

**Результаты работы:**



**Рисунок 5.** Результат

**Вывод:** в ходе выполнения лабораторной работы были получены навыки создания циклов