Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №5

на тему:

**«ЦЕЛОЧИСЛЕННЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ. ОБРАБОТКА МАССИВОВ ЧИСЛОВЫХ ДАННЫХ»**

БГУИР 1-40 04 01

|  |
| --- |
| Выполнил студент группы 253503  ЯРМАК Вероника Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент кафедры информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2023

**Цель работы:** Задание 1. Вариант 8. Ввести массив целых чисел размерностью 30 элементов. Нормализовать значения элементов по максимальному числу в массиве (каждый элемент массива разделить на число с максимальным значением).

**Ход работы:** на листинге 1 представлена программа для решения данной задачи.

Листинг 1 – Исходный код программы задания 1

.model small

.stack 100h

.data

MaxArrayLength equ 30

MaxArrayLength\_x\_2 equ 60

ArrayLength db ?

InputArrayLengthMsqStr db 0Dh, 'Input array length: $'

ErrorInputMsqStr db 0Dh, 'Incorrect value!', 0Ah, '$'

ErrorInputArrayLengthMsqStr db 0Dh, 'Incorrect array length', 0Ah, '$'

InputMsqStr db 0Dh, 'Input: '

Current db 2 dup(0)

InputMsqStrEnding db ' element (-127 .. 127) $'

enterStr db 0Ah, 0Dh, '$'

Answer db 2 dup(0)

ResultMsqStr db 0Dh, 'Result: $'

OstatokMsqStr db 0Dh, ', ostatok $'

Buffer db ?

MaxNumLen db 5

Len db ?

Max db ?

buff db 5 dup (0)

minus db 0

Array db MaxArrayLength dup (0)

result db MaxArrayLength dup(0)

resultArrayLength db ?

.code

start:

mov ax, @data

mov ds, ax

mov es, ax

xor ax, ax

call input

mov ah, 4ch

int 21h

;call Do

;call output

input proc

call inputArrayLength

call InputArray

call FoundMax

call divide

call ShowResult

;mov ah, 4ch

;int 21h

ret

endp

inputArrayLength proc

mov cx, 1

inputArrayLengthLoop:

Call ShowInputArrayLengthMsq

Call inputElementBuff

test ah, ah

jnz inputArrayLengthLoop

cmp Buffer, MaxArrayLength

jg inputArrayLengthLoop\_FAIL

cmp Buffer, 0

jg inputArrayLengthLoop\_OK

inputArrayLengthLoop\_FAIL:

call ShowErrorInputArrayLengthMsqStr

jmp inputArrayLengthLoop

inputArrayLengthLoop\_OK:

mov bl, Buffer

mov ArrayLength, bl

loop inputArrayLengthLoop

ret

endp

ShowInputArrayLengthMsq proc

push ax

push dx

mov ah, 09h

mov dx, offset InputArrayLengthMsqStr

int 21h

pop ax

pop dx

ret

endp

ShowErrorInputArrayLengthMsqStr proc

push ax

push dx

Lea dx, ErrorInputArrayLEngthMsqStr

mov ah, 09h

int 21h

pop ax

pop dx

ret

endp

ShowErrorInputMsq proc

lea dx, ErrorInputMsqStr

mov ah, 09h

int 21h

ret

endp

resetBuffer proc

mov buffer, 0

ret

endp

InputElementBuff proc

push cx

inputElMain:

call resetBuffer

mov ah, 0Ah

mov dx, offset MaxNumLen

int 21h

;mov al, Len

mov dl, 10 ; input '\n' and display

mov ah, 2

int 21h

cmp Len, 0

je errInputEl

mov minus, 0

xor bx, bx

mov bl, Len

lea si, Len

add si, bx

mov bl, 1

xor cx, cx

mov cl, Len

inputElLoop:

std

lodsb; read elements from the rigth to the left

call checkSym

cmp ah, 1

je errInputEl

cmp ah, 2

je nextSym

sub al, '0'

mul bl

test ah, ah

jnz errInputEl

add Buffer, al

jo errInputEl

js errInputEl

mov al, bl

mov bl, 10

mul bl

test ah, ah

jz ElNextCheck

cmp ah, 3 ;;;;;

jne errInputEL

ElNExtCheck:

mov bl, al

jmp nextSym

errInputEl:

call ShowErrorInputMsq

jmp exitInputEl

nextSym:

xor ah, ah

loop inputElLoop

cmp minus, 0

je exitInputEl

neg Buffer

exitInputEl:

pop cx

ret

endp

checkSym proc

cmp al, '-'

je minusSym

cmp al, '9'

ja errCheckSym

cmp al, '0'

jb errCheckSym

jmp exitCheckGood

minusSym:

cmp si, offset Len

je exitWithMinus

errCheckSym:

mov ah, 1

jmp exitCheckSym

exitWithMinus:

mov ah, 2

mov minus, 1

cmp Len, 1

je errCheckSym

jmp exitCheckSym

exitCheckGood:

xor ah, ah

exitCheckSym:

ret

endp

inputArray proc

xor di, di

mov cl, ArrayLength

inputArrayLoop:

call ShowInputMsq

call inputElementBuff

test ah, ah

jnz inputArrayLoop

mov bl, Buffer

mov Array[di], bl

inc di

loop inputArrayLoop

ret

endp

ShowInputMsq proc

mov ax, di ;di contains index

mov bl, 10

div bl

push di

xor di, di

inc di

mov Current[di], ah

add Current[di], '0'

test al, al

jz lessThanTen

dec di

mov Current[di], al

add Current[di], '0'

lessThanTen:

mov ah, 09h

lea dx, InputMsqStr

int 21h

pop di

ret

endp

FoundMax proc

xor dx,dx

xor si, si

xor cx, cx

lea si, Array

mov dl, [Array]

mov cl, ArrayLength

findMaxLoop:

inc si

cmp dl, [si] ; cmp curr with max

jge skipUpdate

mov dl, [si]

skipUpdate:

loop findMaxLoop

mov Max, dl

;mov al, Max

ret

endp

divide proc

lea si, Array

xor di, di

mov cl, ArrayLength

divideLoop:

xor dx, dx

xor ax, ax

mov al, [si]

idiv Max

mov result[di], al

inc di

mov result[di], ah

inc si

inc di

loop divideLoop

xor si, si

xor ax,ax

mov cl, ArrayLength

mov al, 2

mul cl

mov resultArrayLength, al

ret

endp

ShowResult proc

xor si, si

mov cl, resultArrayLength

call PrintResultArray

ret

endp

printResultArray proc

mov ah, 09h

lea dx, ResultMsqStr

int 21h

lea dx, enterStr

mov ah, 09h

int 21h

mov ah, 02h

printLoop:

mov al, result[si]

cmp al, 10

jl lessThanTen1

mov ah, 0

mov bl, 10

div bl

add ah, '0'

mov dl, ah

mov ah, 02h

int 21h

mov al, result[si]

mov ah,0

div bl

add ah, '0'

mov dl, ah

mov ah, 02h

int 21h

jmp printOstatok

lessThanTen1:

;lea dx, OstatokMsqStr

;mov ah, 09h

;int 21h

add al, '0'

mov dl, al

mov ah, 02h

int 21h

printOstatok:

mov dl, 32

mov ah, 02h

int 21h

mov dl, 40

mov ah, 02h

int 21h

mov dl, 111

mov ah, 02h

int 21h

mov dl, 115

mov ah, 02h

int 21h

mov dl, 116

mov ah, 02h

int 21h

mov dl, 32

mov ah, 02h

int 21h

inc si

mov al, result[si]

cmp al, 10

jl lessThanTen2

mov ah, 0

mov bl, 10

div bl

add ah, '0'

mov dl, ah

mov ah, 02h

int 21h

mov al, result[si]

mov ah,0

div bl

add ah, '0'

mov dl, ah

mov ah, 02h

int 21h

jmp printOstatok2

lessThanTen2:

add al, '0'

mov dl, al

mov ah, 02h

int 21h

printOstatok2:

mov dl, 41

mov ah, 02h

int 21h

inc si

dec cl

lea dx, enterStr

mov ah, 09h

int 21h

loop printLoop

ret

endp

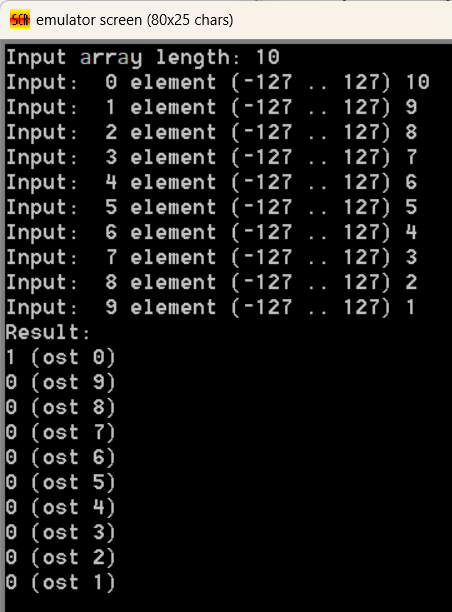


Рисунок 1. Результат выполнения задания 1

**Выводы:** в результате лабораторной работы была выполнена одна задача.