Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**Лабораторная работа №3 (вариант 11)**

по дисциплине: «Ассемблеры и их применение».

Выполнил:

студент 3 курса, гр. ИВТАПбд-31

Кондратьев Павел Сергеевич.

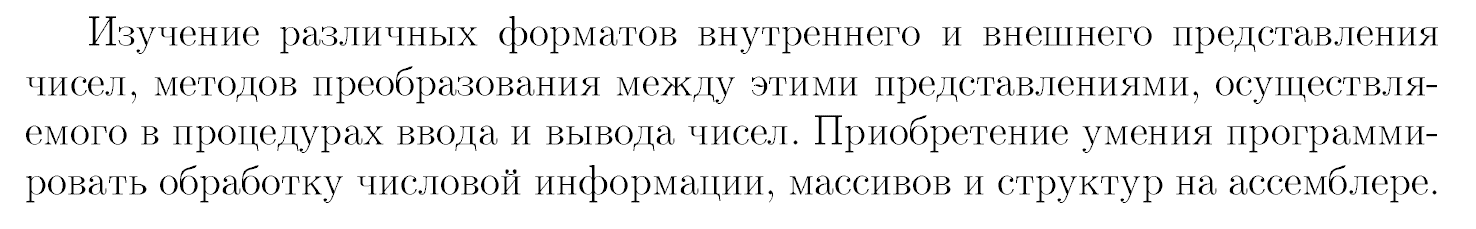
Проверил:

преподаватель кафедры ВТ

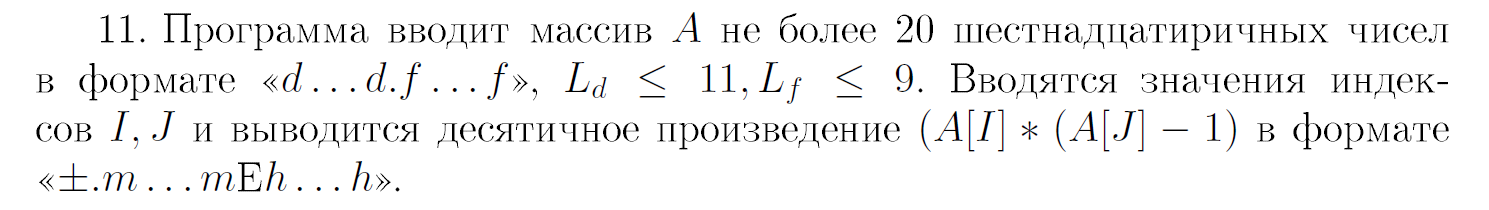
Лылова Анна Вячеславовна.

г. Ульяновск, 2019

**1) Задание:**



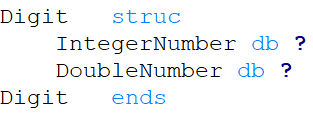
**Задание по варианту**



Задание было обговорено с преподавателем и были внесены изменения:

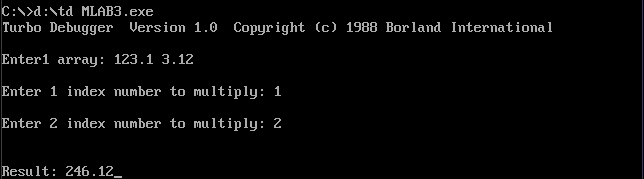
1. Ввод производиться в 10-сс
2. Вывод производиться в 10-сс
3. Выбран формат фиксированной запятой
4. Валидация для вводимого числа
5. Валидация для индексов массива структур

**Описание структуры заданного по заданию**



**2) Контрольные примеры**

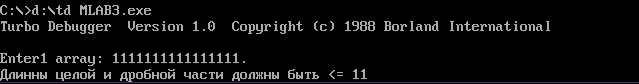
1. Пример работы с двумя элементами в массиве структур



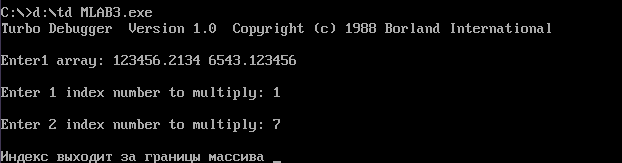
1. Пример работы с несколькими элементами в массиве структур

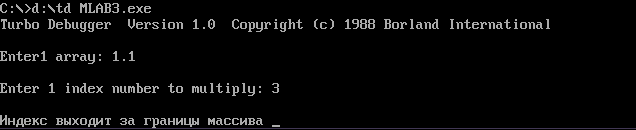


1. Пример ошибки для считывания



1. Пример ошибки для считывания индексов массива





**Код программы:**

.MODEL SMALL

.STACK 200h

.386

Digit struc

IntegerNumber db ?

DoubleNumber db ?

Digit ends

; Декларации данных

.DATA

InputText db 13, 10, "Enter1 array: $"

ForIndex1 db 13, 10, "Enter 1 index number to multiply: $"

ForIndex2 db 13, 10, "Enter 2 index number to multiply: $"Result db 13, 10, "Result: $"

Errors db 13, 10, "Индекс выходит за границы массива $"

Erlen db 13, 10, "Длинны целой и дробной части должны быть <= 11 $"

array Digit 20 dup (<>)

point db ?

i db ?

j db ?

len db ?

arraysize db ?

NL db 13,10,'$'

buff db ?

place db ?

numbint1 dd ?

numbint2 dd ?

numbdouble1 dd ?

numbdouble2 dd ?

;========================= Программа =========================

.CODE

BEGIN LABEL NEAR

; инициализация сегментного регистра

mov AX, @DATA

mov DS, AX

mov ah,9

mov dx,offset InputText

int 21h

mov len, 0

mov point, 0

mov arraysize, 0

mov cl, 0

mov bx,type Digit

mov di, 0

mov si, 0

lea cx, array[di].IntegerNumber

input:

mov ah, 01h

int 21h

cmp al, 20h

je getout

cmp al, 0dh

je way

cmp point, 1

jne next

jmp n2

next:

; запись в структуру

cmp al, '.'

jne n1

cmp len, 11

ja errlen1

jmp Nerr1

errlen1:

mov ah,9

mov dx,offset Erlen

int 21h

jmp ToEnd

Nerr1:

mov len, 0

add point, 1

mov bx, cx

mov [bx], '.'

;lea cx, array[di].DoubleNumber

inc cx

jmp input

n1:

push bx

mov bx, cx

mov [bx], al

inc bx

mov cx, bx

pop bx

inc si

add len, 1

jmp input

n2:

push bx

mov bx, cx

mov [bx], al

inc bx

mov cx, bx

pop bx

inc si

add len, 1

jmp input

getout:

cmp len, 11

ja errlen2

jmp Nerr2

errlen2:

mov ah,9

mov dx,offset Erlen

int 21h

jmp ToEnd

Nerr2:

mov point, 0

mov bx, cx

mov [bx], ' '

inc cx

add di,bx ;к следующей структуре в массиве

add arraysize, 1

jmp input

way:

cmp arraysize, 0

jne step

mov point, 0

add arraysize, 1

step:

add arraysize, 1

mov bx, cx

mov [bx], '/'

mov ah,9

mov dx,offset ForIndex1

int 21h

inputI:

mov ah, 01h

int 21h

cmp al, 0dh

je nextind

; запись 1 индекса

shl i, 2

sub al, 30h

mov i, al

jmp inputI

nextind:

cmp i, 11

ja error

mov dl, i

cmp dl, arraysize

ja error

mov ah,9

mov dx,offset ForIndex2

int 21h

inputJ:

mov ah, 01h

int 21h

cmp al, 0dh

je thatAll

; запись 2 индекса

shl j, 2

sub al, 30h

mov j, al

jmp inputJ

thatAll:

cmp j, 11

ja error

mov dl, j

cmp dl, arraysize

ja error

jmp exit

error:

mov ah,9

mov dx,offset Errors

int 21h

jmp ToEnd

exit:

call Newline

; поиск 2 числе -> лежат в стэке

; верзний элемент стэка -> дробная часть 2 числа

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov bx,type Digit

mov di, 0

mov si, 0

lea bx, array[di].IntegerNumber

mov point, 1

xor dx, dx

mov dl, point

mov cx, 0

mov al, [bx]

cycle:

cmp al, 2fh

je ToEnd

Crep:

cmp dl, i

je Cint

cmp dl, j

je Cint

mov al, [bx]

inc bx

cmp al, ' '

jne Crep

inc dl

jmp Crep

;

; Дописать сбор числа

;

Cint:

mov al, [bx]

cmp al, '.'

je Cnext

push dx

mov dx, 0

sub al, 30h

mov ah, 0

add ax, cx

push ax

mov al, [bx + 1]

cmp al, 2Eh

je qr

pop ax

mov dl, 10

mul dx

mov cx, ax

inc bx

pop dx

jmp Cint

qr:

pop ax

mov cx, ax

inc bx

pop dx

jmp Cint

Cnext:

push ecx

xor cx, cx

inc bx

mov al, [bx]

Cdouble:

mov al, [bx]

cmp al, 2fh

je Cnumb

cmp al, ' '

je Cnumb

push dx

mov dx, 0

sub al, 30h

mov ah, 0

add ax, cx

push ax

mov al, [bx + 1]

cmp al, ' '

je qr2

cmp al, 2fh

je qr2

pop ax

mov dl, 10

mul dx

mov cx, ax

inc bx

pop dx

jmp Cdouble

qr2:

pop ax

mov cx, ax

inc bx

pop dx

jmp Cdouble

Cnumb:

push ecx

xor cx, cx

inc bx

inc dx

jmp cycle

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;; Вывод ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov bx,type Digit

mov di, 0

mov si, 0

lea cx, array[di].IntegerNumber

mov bx, cx

start:

cmp [bx], '/'

je LoopOut

mov ah, 02h

mov dl,[bx]

int 21h

inc bx

jmp start

LoopOut:

jmp ToEnd

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;; Вывод числа ;;;;;;;;;;;;;;;;;;;;;

print:

mov eax, edx

xor ecx, ecx

mov ebx, 10

oi2:

xor edx,edx

div ebx

push edx

inc ecx

test eax, eax

jnz oi2

mov ah, 02h

oi3:

pop edx

add dl, '0'

int 21h

loop oi3

ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;; перенос троки ;;;;;;;;;;;;;;;;

Newline proc

push ax

push dx

mov ah,9

lea dx,NL

int 21h

pop dx

pop ax

ret

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

ToEnd:

pop ecx

mov numbdouble2, ecx

pop ecx

mov numbint2, ecx

pop ecx

mov numbdouble1, ecx

pop ecx

mov numbint1, ecx

mov ah,9

mov dx,offset Result

int 21h

mov eax, numbint1

mov ebx, numbint2

sub ebx, 1

mul ebx

mov edx, eax

call print

mov dl, '.'

int 21h

mov eax, numbdouble1

mov ebx, numbdouble2

mul ebx

mov edx, eax

call print

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ax, 4c00h

int 21h

END BEGIN

**Вывод:**

В ходе выполнения данной лабораторной работы, я изучил различные форматы внутреннего и внешнего представления чисел (фиксированная точка), методов преобразования осуществляемого в процедурах ввода и вывода чисел. Приобретение умения программировать обработку числовой информации, массивов и структур на ассемблере.

**Список литературы:**

* Вострикова З. П. Программирование на языке ассемблера ЕС ЭВМ. М.: Наука, 1985.
* Калашников О. А. Ассемблер? Это просто! Учимся программировать. — [БХВ-Петербург](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3&action=edit&redlink=1), 2011.
* Юров В., Хорошенко С. Assembler: учебный курс. — СПб.: [Питер](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE))