



**«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

**О Т Ч Е Т**

по **лабораторной работе** № 4

Дисциплина: Машинно-зависимые языки и основы компиляции

Название **лабораторной работы**: Программирование обработки

массивов и матриц

Студент гр. ИУ6-42Б \_\_\_\_\_ Медведев АЕ  
(Подпись, дата) (И.О. Фамилия)

Преподаватель \_\_\_\_\_  
(Подпись, дата) (И.О. Фамилия)

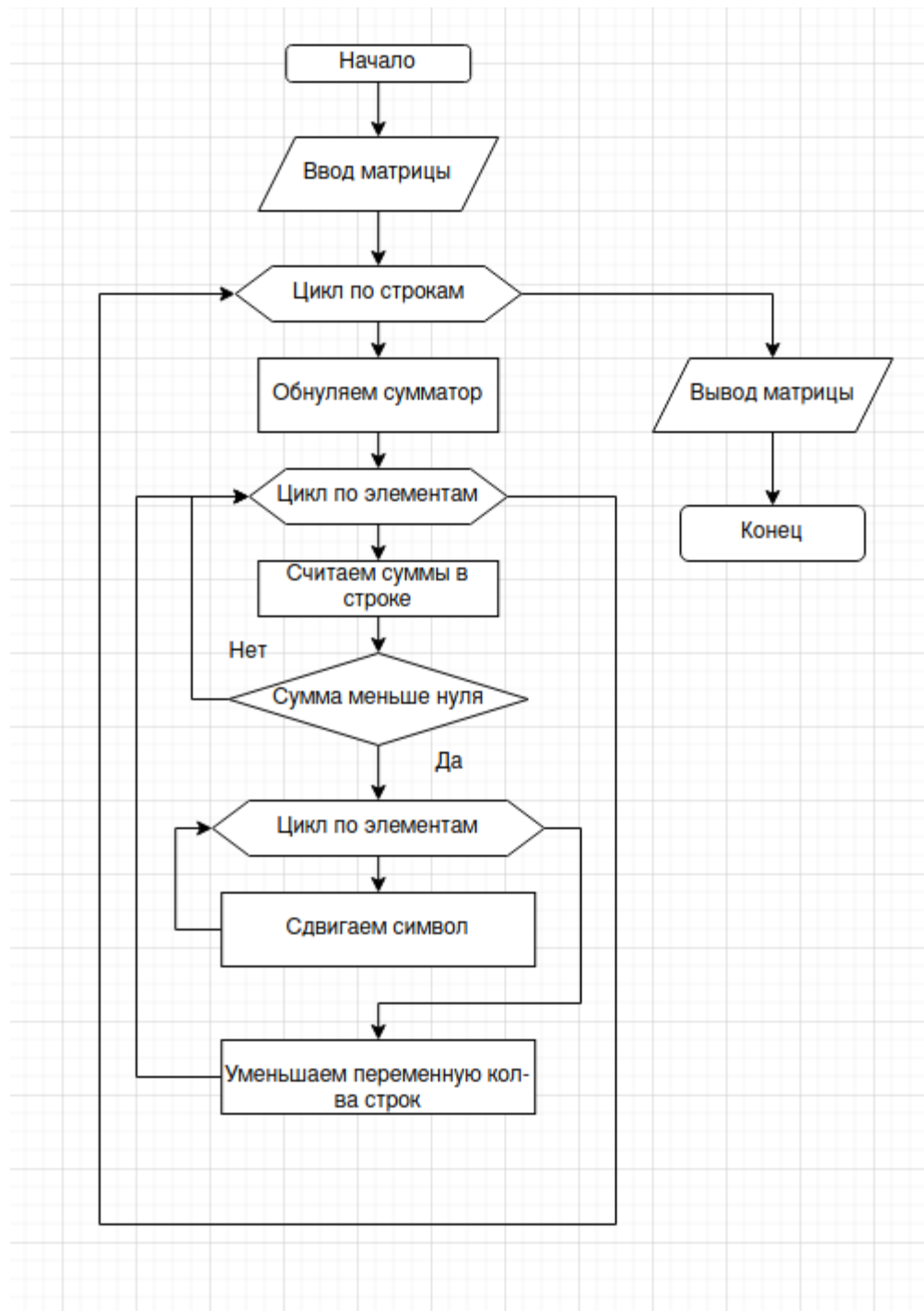
Москва, 2021\_

**Цель работы:** изучение приемов моделирования обработки массивов и матриц в языке ассемблера.

**Задание:**

**Лабораторная работа №4. Программирование обработки массивов и матриц.**

Дана матрица 6x4. Вычеркнуть строки с отрицательной суммой элементов. Организовать ввод матрицы и вывод результатов.



(рис.1 схема алгоритма)

**Код:**

```
; 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 "Press Enter to Exit",0AH,0DH,0
MsgInput db "Input 24 value for array 4 * 6", 0ah, 0dh, 0
endline db 0ah, 0dh, 0
space db " ", 0

.DATA
;array dd 4 dup (2)
; dd 4 dup (-2)
; dd 4 dup (5)
; dd 4 dup (23)
; dd 4 dup (-15)
; dd 4 dup (39)
w dw 4
h dw 6
.DATA?
array dd 24 dup(?)
inbuf DB 100 DUP (?)
output db 100 dup (?)
buffer db 100 dup (?)
bword db 20 dup(?)
.CODE
Start:

; input array
Invoke StdOut, ADDR MsgInput
mov EBX, 0

cin:
cmp ebx, 24
je cend
Invoke StdIn, ADDR buffer, LengthOf buffer
Invoke StripLF, ADDR buffer
```

```

    Invoke atol,  ADDR buffer
    cmp eax, 0h
    je cin
    MOV  array[EBX*4], EAX
    inc  EBX
    cmp ebx, 24
    je cend

```

nextint:

```

    Invoke atol,  edx
    cmp eax, 0h
    je cin
    MOV  array[EBX*4], EAX
    inc  EBX
    cmp ebx, 24
    je cend
    jmp nextint

```

cend:

```

    xor ecx, ecx
    mov cx, h
    xor ebx, ebx
    lea edi, array
    lea esi, array
for_i:
    push ecx
    mov cx, w
    xor eax, eax
    xor ebx, ebx
for_j:
    add eax, [edi + ebx * 4]
    inc ebx
    loop for_j

    cmp eax, 0; check the sum
    jl nenorm
    xor ebx, ebx
    mov cx, w
for_offset:
    mov eax, [edi + ebx * 4]
    mov [esi + ebx * 4], eax
    inc ebx
    loop for_offset
    xor eax, eax
    xor ebx, ebx

```

```

        mov ax, w
        mov bx, 4
        mul bx
        add esi, eax
        jmp norm
nenorm:
        mov ax, h
        dec ax
        mov h, ax
norm:
        xor eax, eax
        xor ebx, ebx
        mov ax, w
        mov bx, 4
        mul bx
        add edi, eax
        pop ecx
        loop for_i

        ;print
        xor ecx, ecx
        mov cx, h
        xor ebx, ebx
        XOR    EAX,EAX
        lea edi, array
for_i_print:
        push ecx
        mov cx, w
for_j_print:
        push ecx
        Invoke dwtoa, [edi + ebx * 4], ADDR output
        inc ebx
        Invoke StdOut,ADDR output
        Invoke StdOut,ADDR space
        pop ecx
        loop for_j_print
        Invoke StdOut,ADDR endlne
        pop ecx
        loop for_i_print

        XOR    EAX,EAX
        Invoke StdOut,ADDR MsgExit
        Invoke StdIn,ADDR inbuf,LengthOf inbuf

        Invoke ExitProcess,0
        End    Start

```

```

C:\RADASM\Masm\Projects\lab4\lab4.exe
Input 24 value for array 4 * 6
1 1 1 1
-4 1 1 1
2 2 2 2
-6 1 1 1
3 3 3 3
5 5 5 5
1 1 1 1
2 2 2 2
3 3 3 3
5 5 5 5
Press Enter to Exit

```

(рис. 2 работа программы, тест 1)

```

Выбрать C:\RADASM\Masm\Projects\lab4\lab4.exe
Input 24 value for array 4 * 6
1 1 1 1
9 9 9 9
8 8 8 8
7 7 7 7
6 6 6 6
5 5 5 5
1 1 1 1
9 9 9 9
8 8 8 8
7 7 7 7
6 6 6 6
5 5 5 5
Press Enter to Exit

```

(рис. 4 работа программы, тест 2)

```

Выбрать C:\RADASM\Masm\Projects\lab4
Input 24 value for array 4 * 6
1 3 1 1
-9 1 3 2
-7 -4 2 1
3 3 1 5
6 4 1
3
4 2 1 4
1 3 1 1
3 3 1 5
6 4 1 3
4 2 1 4
Press Enter to Exit

```

(рис. 4 работа программы, тест 3)

Ввод	Ожидалось	Вывод
1 1 1 1 -4 1 1 1 2 2 2 2 -6 1 1 1 3 3 3 3 5 5 5 5	1 1 1 1 2 2 2 2 3 3 3 3 5 5 5 5	1 1 1 1 2 2 2 2 3 3 3 3 5 5 5 5
1 1 1 1 9 9 9 9 8 8 8 8 7 7 7 7 6 6 6 6 5 5 5 5	1 1 1 1 9 9 9 9 8 8 8 8 7 7 7 7 6 6 6 6 5 5 5 5	1 1 1 1 9 9 9 9 8 8 8 8 7 7 7 7 6 6 6 6 5 5 5 5
1 3 1 1	1 3 1 1	1 3 1 1

-9 1 3 2	3 3 1 5	3 3 1 5
-7 -4 2 1	6 4 1 3	6 4 1 3
3 3 1 5	4 2 1 4	4 2 1 4
6 4 1 3		
4 2 1 4		

### Контрольные вопросы

1. Почему в ассемблере не определены понятия «массив», «матрица»?

Потому что массив и матрица — это последовательность данных, с которыми можно работать не прибегая к усложнению языка

2. Как в ассемблере моделируются массивы?

Как последовательность байт, слов, двойных слов.

3. Поясните фрагмент последовательной адресации элементов массива?

Почему при этом для хранения частей адреса используют регистры?

Нельзя производить операции между памятью и памятью, один из операндов всегда должен быть в регистре поэтому элементы в программе последовательно выгружаются в регистры

4. Как в памяти компьютера размещаются элементы матриц?

Для статической матрицы элементы размещаются последовательно, как для одномерного массива такого же размера.

5. Чем моделирование матриц отличается от моделирования массивов? В

каких случаях при выполнении операций для адресации матриц используется один регистр, а в каких – два?

В матрицах следует различать просмотр по строкам, просмотр по столбцам, просмотр по диагоналям и произвольный доступ. Если матрица расположена в памяти по строкам и просмотр выполняется по строкам, то обработка может выполняться так, как в одномерном массиве, без учета перехода от одной строки к другой.

Просмотр по строкам при необходимости фиксировать завершение строки и просмотр по столбцам при построчном расположении в памяти выполняются в двойном цикле. При просмотре по диагонали обычно используют один цикл, через переменную которого рассчитываются смещения элементов массива. Однако проще использовать специальный регистр смещения, который должен соответствующим образом переадресовываться.

**Вывод:** В ходе лабораторной работы была создана программа на языке ассемблера, которая удаляет строки в матрице, сумма элементов которой отрицательна.