



**«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

**О Т Ч Е Т**

**по лабораторной работе № 4**

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

**Название лабораторной работы: Обработки массивов и матриц**

Студент гр. ИУ6-41Б \_\_\_\_\_  
(Подпись, дата) (И.О. Фамилия)

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

Москва, 2025

## СОДЕРЖАНИЕ

Цель работы .....	3
Задание .....	3
Схема алгоритма.....	3
Реализация программы .....	4
Тестирование программы.....	9
Вывод.....	11
Контрольные вопросы .....	12

## Цель работы

Изучение приемов моделирования обработки массивов и матриц в языке ассемблера.

## Задание

Дана матрица 6x5. Вычеркнуть столбец с заданным номером. Организовать ввод матрицы и вывод результатов.

## Схема алгоритма

Разработаем схему алгоритма реализуемой программы. Результат выполнения этого пункта представлен на рисунке 1.

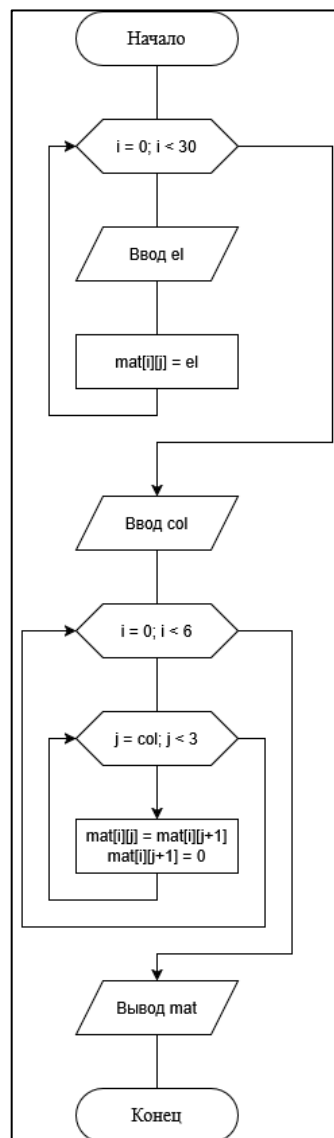


Рисунок 1 — Схема алгоритма

## Реализация программы

Программа реализована на языке ассемблера. Результат представлен на листинге 1.

### Листинг 1 — Код программы

```
section .data
    matrix times 30 dd 0x20202020
    input_msg db "Input matrix (element by line)", 10
    len_input_msg equ $-input_msg
    input_el_msg db "[1][1] = "
    len_input_el_msg equ $-input_el_msg
    first_m_msg db "Your matrix:", 10
    len_first_m_msg equ $-first_m_msg
    col_del_msg db "Input column to delete: "
    len_col_del_msg equ $-col_del_msg
    col_del db 0
    newline db 10
    len_newline equ $-newline
    inbuf dd 0x20202020
    len_inbuf equ $-inbuf
section .bss

section .text
    global _start
_start:

end_program:
    mov eax, 4
    mov ebx, 1
    mov ecx, input_msg
    mov edx, len_input_msg
    int 80h
```

## Продолжение листинга 1

```
    mov ecx, 30
    mov esi, matrix
input_loop:
    push ecx
    mov dword[inbuf], 0x20202020
    call print_msg_el
    mov eax, 3
    mov ebx, 0
    mov ecx, inbuf
    mov edx, len_inbuf
    int 80h
    call remove_nextline

    call inc_el
    add esi, 4
    pop ecx
    loop input_loop
    mov eax, 4
    mov ebx, 1
    mov ecx, first_m_msg
    mov edx, len_first_m_msg
    int 80h
    mov ecx, 6
    mov esi, matrix
print_first_matrix:
    push ecx
    mov eax, 4
    mov ebx, 1
    mov ecx, esi
    mov edx, 20
    int 80h
    mov eax, 4
```

## Продолжение листинга 1

```
    mov ebx, 1
    mov ecx, newline
    mov edx, 1
    int 80h
    add esi, 20
    pop ecx
    loop print_first_matrix

    mov eax, 4
    mov ebx, 1
    mov ecx, col_del_msg
    mov edx, len_col_del_msg
    int 80h

    mov eax, 3
    mov ebx, 0
    mov ecx, col_del
    mov edx, 1
    int 80h

    movzx eax, byte [col_del]
    sub eax, '1'
    mov [col_del], al

    mov ecx, 6
    mov esi, matrix

del_col:
    push ecx
    movzx edi, byte [col_del]
    mov edx, esi
```

## Продолжение листинга 1

```
    mov ebx, edi
shift_loop:
    cmp ebx, 4
    jge shift_done
    mov eax, [edx + ebx*4 + 4]
    mov [edx + ebx*4], eax
    inc ebx
    jmp shift_loop

shift_done:
    mov dword [edx + 4*4], 0x20202020

    add esi, 20
    pop ecx
    loop del_col

    mov eax, 4
    mov ebx, 1
    mov ecx, first_m_msg
    mov edx, len_first_m_msg
    int 80h

    mov ecx, 6
    mov esi, matrix
print_modified_matrix:
    push ecx
    mov eax, 4
    mov ebx, 1
    mov ecx, esi
    mov edx, 16
    int 80h
```

## Продолжение Листинга 1

```
    mov eax, 4
    mov ebx, 1
    mov ecx, newline
    mov edx, 1
    int 80h

    add esi, 20
    pop ecx
    loop print_modified_matrix

    mov eax, 1
    xor ebx, ebx
    int 80h

exit_program:
    mov eax, 1
    mov ebx, 0
    int 80h
print_msg_el:
    mov eax, 4
    mov ebx, 1
    mov ecx, input_el_msg
    mov edx, len_input_el_msg
    int 80h
    ret
inc_el:
    add byte[input_el_msg + 4], 1
    mov eax, [input_el_msg + 4]
    cmp al, "6"
    je el_next_str
    ret
el_next_str:
```



## Продолжение Листинга 1

```
    mov al, "1"
    mov byte[input_el_msg + 4], al
    add byte[input_el_msg + 1], 1
    ret
remove_nextline:
    cmp byte[inbuf], 10
    je end_rem_nl
    mov al, byte[inbuf]
    mov byte[esi], al
    cmp byte[inbuf+1], 10
    je end_rem_nl
    mov al, byte[inbuf+1]
    mov byte[esi+1], al
    cmp byte[inbuf+2], 10
    je end_rem_nl
    mov al, byte[inbuf+2]
    mov byte[esi+2], al
    cmp byte[inbuf+3], 10
    je end_rem_nl
    mov al, byte[inbuf+3]
    mov byte[esi+3], al
end_rem_nl:
    ret
```

## Тестирование программы

Результаты тестирования программы на трёх наборах данных.  
Результаты тестирования в таблице 1.

Таблица 1 — результаты тестирования.

Исходные данные	Ожидаемый результат	Полученный результат
1 2 3 4 5	2 3 4 5	2 3 4 5
2 3 4 5 6	3 4 5 6	3 4 5 6
3 4 5 6 7	4 5 6 7	4 5 6 7
4 5 6 7 8	5 6 7 8	5 6 7 8
5 6 7 8 9	6 7 8 9	6 7 8 9
6 7 8 9 0	7 8 9 0	7 8 9 0
1		

## **Вывод**

Отработаны приемы обработки массивов и реализации вложенных циклов на языке ассемблер. Реализована программа, которая проводит операции с матрицей.

## Контрольные вопросы

1. В ассемблере не определены понятия массив и матрица потому что ассемблер является низкоуровневым языком программирования, который оперирует непосредственно с регистрами процессора, памятью и инструкциями процессора. В ассемблере отсутствуют высокоуровневые абстракции данных, которые присущи другим языкам программирования более высокого уровня.

2. В ассемблере для моделирования массивов обычно используется работа с памятью и адресацией. Это означает, что каждый элемент массива будет храниться по определенному адресу в памяти.

3. Для последовательной адресации элементов массива в языке ассемблера необходимо использовать схему база + смещение. В одном регистре необходимо хранить адрес первого элемента массива и прибавлять к нему номер элемента к которому мы хотим добраться. Необходимо следить за размером смещения и эл-тов массива.

4. Элементы матрицы в памяти компьютера обычно размещаются последовательно, по строкам.

5. Моделирование матриц отличается от моделирования массивов прежде всего тем, что матрица представляет собой двумерную структуру данных, тогда как массив – одномерная. Два регистра может понадобиться, если проходить матрицу не последовательно, а, например, по диагонали.