

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

МИКРОПРОЕКТ №1
ПРОГРАММА ДЛЯ УМНОЖЕНИЯ МАТРИЦ ПОРЯДКА 5
НА NASM

Пояснительная записка

Инв. №подл	Подп. и дата	Инв. № дубл.	Подп. и дата

Исполнитель:

Студент группы БПИ191_1,
Соколовский Вацлав Антонович

Москва 2020

1. ВВЕДЕНИЕ4
 - 1.1. Наименование программы.....**Ошибка! Закладка не определена.**
 - 1.2. Документы, на основании которых ведется разработка... **Ошибка! Закладка не определена.**
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ..... **ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.**
 - 2.1. Назначение программы.....**Ошибка! Закладка не определена.**
 - 2.1.1. Функциональное назначение.....**Ошибка! Закладка не определена.**
 - 2.1.2. Эксплуатационное назначение.....**Ошибка! Закладка не определена.**
 - 2.2. Краткая характеристика области применения..... **Ошибка! Закладка не определена.**
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ **ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.**
 - 3.1. Постановка задачи на разработку программы..... **Ошибка! Закладка не определена.**
 - 3.2. Описание используемого математического аппарата.. **Ошибка! Закладка не определена.**
 - 3.2.1. Нейронная сеть прямого распространения (Feed Forward Neural Network) **Ошибка! Закладка не определена.**
 - 3.2.2. Гиперпараметры**Ошибка! Закладка не определена.**
 - 3.2.3. Используемая функция потерь.....**Ошибка! Закладка не определена.**
 - 3.2.4. Используемый оптимизатор**Ошибка! Закладка не определена.**
 - 3.3. Описание алгоритма и функционирования программы ... **Ошибка! Закладка не определена.**
 - 3.3.1. Основной алгоритм функционирования приложения регистрации **Ошибка! Закладка не определена.**
 - 3.3.1.1. Алгоритм добавления пользователя**Ошибка! Закладка не определена.**
 - 3.3.1.2. Алгоритм удаления пользователя.....**Ошибка! Закладка не определена.**
 - 3.3.2. Основной алгоритм функционирования приложения аутентификации **Ошибка! Закладка не определена.**
 - 3.3.2.1. Алгоритм аутентификации**Ошибка! Закладка не определена.**
 - 3.3.3. Алгоритм хеширования изображения по среднему (Average Hash). **Ошибка! Закладка не определена.**
 - 3.4. Обоснование выбора алгоритма решения задачи..... **Ошибка! Закладка не определена.**
 - 3.4.1. Возможные взаимодействия с другими программами. **Ошибка! Закладка не определена.**
 - 3.5. Описание и обоснование метода организации входных и выходных данных. **Ошибка! Закладка не определена.**
 - 3.5.1. Описание метода организации входных и выходных данных..... **Ошибка! Закладка не определена.**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.5.1.1.	Компонента регистрации	Ошибка! Закладка не определена.
3.5.1.2.	Компонента аутентификации	Ошибка! Закладка не определена.
3.5.2.	Обоснование метода организации входных и выходных данных.	Ошибка! Закладка не определена.
3.6.	Описание и обоснование выбора состава технических и программных средств	Ошибка! Закладка не определена.
4.	ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
4.1.	Предполагаемая потребность	Ошибка! Закладка не определена.
4.2.	Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами	Ошибка! Закладка не определена.
5.	ПЕРСПЕКТИВЫ РАЗВИТИЯ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
ИСТОЧНИКИ:		10
ПРИЛОЖЕНИЕ 1		11
ПРИЛОЖЕНИЕ 2		ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
ПРИЛОЖЕНИЕ 3		ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
ПРИЛОЖЕНИЕ 4		ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

■ Текст задания

Разработать программу умножения матриц порядка $N=5$ при условии размещения элементов матриц в линейных массивах по строкам.

■ Дополнительный реализованный функционал

В программу добавлена возможность умножения квадратных матриц порядка в диапазоне $[1; 32767]$.

■ Расчетные методы

При умножении двух матриц (A и B) использовались самостоятельно выведенные формулы для доступа к элементам $[i, k]$ и $[j, k]$, где i - номер строки в матрице A, j - номер столбца в матрице B, а k - индекс элементов из i -ой строки матрицы A и j -ого столбца матрицы B соответственно.

Формула для доступа к элементу $[i, k]$ ($i, k \in [0, N - 1]$, где N - порядок матрицы):

$$indexOfElementA = i \cdot N + k;$$

Формула для доступа к элементу $[j, k]$ ($j, k \in [0, N - 1]$, где N - порядок матрицы):

$$indexOfElementA = k \cdot N + j;$$

Формула для доступа к элементу результирующей матрицы, куда записывается сумма произведений элементов i -ой строки матрицы A и j -ого столбца матрицы B:

$$indexOfElementA = i \cdot N + j;$$

Сама программа реализуется с помощью циклов уровня вложенности 3, где в первом цикле идет итерация по строкам матрицы A, во втором цикле - итерация по столбцам матрицы B и запись результата третьего цикла в элемент результирующей матрицы, в третьем цикле - итерация по элементам двух матриц и подсчет суммы их произведений.

■ Параметры ввода и их область значения

- N – порядок матриц для работы. Область значения для корректной работы: $[1; 32767]$
- X – отдельный элемент матриц A и B. Область значения для корректной работы: $[-32768; 32767]$

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. ТЕСТИРОВАНИЕ

Единичные матрицы

```

Please, type size of an matrix's row:
5
Please, type your (int) matrix:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Your new matrix:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

Please, type size of an matrix's row:
5
Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Please, type your (int) matrix:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

Your new matrix:
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

■ Диагональные матрицы

```
Please, type size of an matrix`s row:
5
Please, type your (int) matrix:
2 0 0 0 0
0 2 0 0 0
0 0 2 0 0
0 0 0 2 0
0 0 0 0 2

Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Your new matrix:
2 4 6 8 10
12 14 16 18 20
22 24 26 28 30
32 34 36 38 40
42 44 46 48 50
```

```
Please, type size of an matrix`s row:
5
Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Please, type your (int) matrix:
2 0 0 0 0
0 2 0 0 0
0 0 2 0 0
0 0 0 2 0
0 0 0 0 2

Your new matrix:
2 4 6 8 10
12 14 16 18 20
22 24 26 28 30
32 34 36 38 40
42 44 46 48 50
```

■ Нулевые матрицы

```
Please, type size of an matrix`s row:
5
Please, type your (int) matrix:
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Your new matrix:
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

Please, type size of an matrix's row:
5
Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Please, type your (int) matrix:
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

Your new matrix:
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

```

Обычные матрицы

```

Please, type size of an matrix's row:
5
Please, type your (int) matrix:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Please, type your (int) matrix:
25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Your new matrix:
175 160 145 130 115
550 510 470 430 390
925 860 795 730 665
1300 1210 1120 1030 940
1675 1560 1445 1330 1215

```

Проверка:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix} \cdot \begin{pmatrix} 25 & 24 & 23 & 22 & 21 \\ 20 & 19 & 18 & 17 & 16 \\ 15 & 14 & 13 & 12 & 11 \\ 10 & 9 & 8 & 7 & 6 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 175 & 160 & 145 & 130 & 115 \\ 550 & 510 & 470 & 430 & 390 \\ 925 & 860 & 795 & 730 & 665 \\ 1300 & 1210 & 1120 & 1030 & 940 \\ 1675 & 1560 & 1445 & 1330 & 1215 \end{pmatrix}$$

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Обратные матрицы

```

Please, type size of an matrix`s row:
5
Please, type your (int) matrix:
1 2 3 4 5 0 1 2 3 4 0 0 1 2 3 0 0 0 1 2 0 0 0 0 1

Please, type your (int) matrix:
1 -2 1 0 0 0 1 -2 1 0 0 0 1 -2 1 0 0 0 1 -2 0 0 0 0 1

Your new matrix:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

```

```

Please, type size of an matrix`s row:
5
Please, type your (int) matrix:
1 -2 1 0 0 0 1 -2 1 0 0 0 1 -2 1 0 0 0 1 -2 0 0 0 0 1

Please, type your (int) matrix:
1 2 3 4 5 0 1 2 3 4 0 0 1 2 3 0 0 0 1 2 0 0 0 0 1

Your new matrix:
1 0 0 0 0
0 1 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1

```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Большие числа

```

Please, type size of an matrix's row:
5
Please, type your (int) matrix:
32767 32767 32767 32767 32767
-32768 -32768 -32768 -32768 -32768
32767 32767 32767 32767 32767
-32768 -32768 -32768 -32768 -32768
32767 32767 32767 32767 32767

Please, type your (int) matrix:
32767 32767 32767 32767 32767
-32768 -32768 -32768 -32768 -32768
32767 32767 32767 32767 32767
-32768 -32768 -32768 -32768 -32768
32767 32767 32767 32767 32767

Your new matrix:
1073610755 1073610755 1073610755 1073610755 1073610755
-1073643520 -1073643520 -1073643520 -1073643520 -1073643520
1073610755 1073610755 1073610755 1073610755 1073610755
-1073643520 -1073643520 -1073643520 -1073643520 -1073643520
1073610755 1073610755 1073610755 1073610755 1073610755

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ИСТОЧНИКИ:

1. NASM Forum [Электронный ресурс] – Режим доступа: <https://forum.nasm.us>, свободный. (дата обращения: 29.10.2020).
2. Размерность типов данных [Электронный ресурс] – Режим доступа: <https://docs.mql4.com/ru/basis/types/integer/integertypes>, свободный. (дата обращения: 29.10.2020).
3. Stackoverflow Forum [Электронный ресурс] – Режим доступа: <https://stackoverflow.com>, свободный. (дата обращения: 29.10.2020).
4. Условные и безусловные переходы в ASM [Электронный ресурс] – Режим доступа: <http://osinavi.ru/asm/4.html>, свободный. (дата обращения: 29.10.2020).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

ТЕКСТ ПРОГРАММЫ

```
; nasm -fmacho64 MatrixMultiplication.asm && gcc MatrixMultiplication.o &&
./a.out

; Разработать программу умножения матриц порядка N=5
; при условии размещения элементов матриц в линейном
; массиве по строкам.

global    _main

extern    _scanf                ; импорт команд из другого модуля
extern    _printf
extern    _malloc
extern    _free

section .data

formatStr db '%s', 0
formatInt db '%d', 0
formatIntWithSpace db '%d ', 0

okMessage db 'Everything is OK.', 10, 13, 0 ; сообщение для дебага :)
typeSize db 'Please, type size of an matrix`s row: ', 10, 13, 0
typeMatrix db 'Please, type your (int) matrix: ', 10, 13, 0
yourMatrix db 'Your new matrix:', 10, 13, 0
wrongInput db 'Incorrect input. Please try again', 10, 13, 10, 13, 0
newLine db 10, 13, 0

N: times 8 db 0                ; порядок матрицы
sqr_N: times 8 db 0            ; размер матрицы
number: times 8 db 0           ; введенное число
temp: times 8 db 0
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

    iterator: times 8 db 0      ; вспомогательная переменная для циклов loop

    i: times 8 db 0             ; счетчик в цикле

    j: times 8 db 0             ; счетчик в цикле уровня 2

    k: times 8 db 0             ; счетчик в цикле уровня 3


    matrix times 8 db 0         ; первая матрица

    matrix2: times 8 db 0       ; вторая матрица

    answer_matrix: times 8 db 0 ; матрица для результата


    one dd 1                    ; константа для сравнения

    zero dd 0


section .text

    _main:

        push    rbp

        mov     rdi, typeSize
        call    _printf

        mov     rdi, formatInt
        mov     rsi, N
        call    _scanf          ; считываем порядок квадратной матрицы


        mov     eax, [rel N]

        cmp     eax, [rel one]  ; проверяем, положителен ли порядок
        jl      _WrongInput


        mov     rax, [rel N]

        mov     rbx, [rel N]

        mul     rbx

        mov     [rel sqr_N], rax

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

mov rdi, typeMatrix

call _printf          ; просим ввести матрицу построчно

mov rdi, [rel sqr_N]

imul rdi, 8

call _malloc          ; ищем участок памяти на N^2 чисел размером
8 байт

mov [rel matrix], rax ; записываем в matrix адрес первой ячейки
найденного участка

call InputMatrix      ; идем в процедуру ввода матрицы

mov rdi, newLine

call _printf

xor r11, r11

mov [rel i], r11      ; обнуляем счетчик цикла

mov rdi, typeMatrix

call _printf          ; просим ввести вторую матрицу построчно

mov rdi, [rel sqr_N]

imul rdi, 8

call _malloc          ; ищем участок памяти на N^2 чисел размером
8 байт

mov [rel matrix2], rax ; записываем в matrix2 адрес первой ячейки
найденного участка

call InputSecondMatrix ; идем в процедуру ввода второй матрицы

mov rdi, newLine

call _printf

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

mov rdi, [rel sqr_N]

imul rdi, 8

call _malloc          ; ищем участок памяти на N^2 чисел размером
8 байт

mov [rel answer_matrix], rax ; записываем в matrix2 адрес первой
ячейки найденного участка

call MultiplyMatrixes

mov rdi, [rel matrix]

call _free

; освобождаем зарезервированную память

mov rdi, [rel matrix2] ; под промежуточные матрицы после создания
результатирующей

call _free

xor r11, r11

mov [rel i], r11      ; обнуляем счетчик цикла

call OutputMatrix

mov rdi, [rel answer_matrix] ; освобождаем память матрицы-
результата после ее вывода

call _free

jmp ExitProgram

; --ПРОЦЕДУРЫ--

InputMatrix:

push rbx

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

mov rcx, [rel sqr_N]

ReadMatrix:

    mov [rel iterator], rcx; махинации с сохранением в
переменную, ибо rcx может меняться в цикле

                                ; подобную магию мы будем делать в
каждом цикле, ибо все они крупные

                                ; и мы не можем быть уверены в
сохранности регистра

    mov rdi, formatInt
    mov rsi, number
    call _scanf

    mov r10, [rel i]
    imul r10, dword 8      ; вычисляем сдвиг указателя

    mov rax, [rel matrix]
    add rax, r10           ; смещаем указатель на только что
посчитанный сдвиг

    mov r10, [rel number] ; запишем введенное число в есх
    mov [rel rax], r10     ; записали введенное число в ячейку
массива

    mov r11, [rel i]
    add r11, 1             ; инкрементируем счетчик
    mov [rel i], r11

    mov rcx, [rel iterator]
    loop ReadMatrix

    mov rax, 1
    pop rbx
    ret

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

InputSecondMatrix:

    push    rbx

    mov     rcx, [rel sqr_N]

    ReadSecondMatrix:

        mov     [rel iterator], rcx; махинации с сохранением в
переменную, ибо rcx может меняться в цикле

        mov     rdi, formatInt
        mov     rsi, number
        call    _scanf

        mov     r10, [rel i]
        imul    r10, dword 8      ; вычисляем сдвиг указателя

        mov     rax, [rel matrix2]
        add     rax, r10          ; смещаем указатель на только что
посчитанный сдвиг

        mov     r10, [rel number] ; запишем введенное число в r10
        mov     [rel rax], r10    ; записали введенное число в ячейку
массива

        mov     r11, [rel i]
        add     r11, 1            ; инкрементируем счетчик
        mov     [rel i], r11

        mov     rcx, [rel iterator]
        loop    ReadSecondMatrix

    mov     rax, 1
    pop     rbx
    ret

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата


```

OutputMatrix:

    push    rbx

    mov     rdi, yourMatrix ; выводим сообщение о том, что дальше последует
вывод матрицы-результата

    call    _printf

    mov     rcx, [rel N]

    ColumnOutputIterator: ; итерация по колонке (перебираем строки)

        mov [rel i], rcx

        mov     rcx, [rel N]

        RowOutputIterator: ; итерация по строкам (перебираем
элементы)

            mov [rel j], rcx

            mov     r10, [rel N] ; вычитаем текущий индекс из
длины для получения индекса элемента

            sub     r10, [rel i] ; i и j принадлежат диапазону
[1;N]

            imul    r10, [rel N]

            add     r10, [rel N]

            sub     r10, [rel j]

            imul    r10, dword 8 ; просчитываем сдвиг указателя

            mov     rax, [rel answer_matrix]

            add     rax, r10 ; сдвигаем указатель к нужному
элементу

            mov     rdi, formatIntWithSpace

            mov     rsi, [rel rax] ; выводим нужный элемент

            call    _printf

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

                                mov rcx, [rel j]

                                loop RowOutputIterator

                                mov rdi, newLine

                                call _printf

                                mov rcx, [rel i]

                                loop ColumnOutputIterator

                                mov rdi, newLine

                                call _printf

                                xor rax, rax

                                pop rbx

                                ret

MultiplyMatrixes:

                                push rbx

                                mov rcx, [rel N]

                                RowsIteration:

                                    mov [rel i], rcx

                                    mov rcx, [rel N]

                                    ColumnIteration:

                                        mov [rel j], rcx

                                        xor r12, r12 ; будем использовать для
хранения суммы произведений

                                        mov rcx, [rel N]
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

ElemIteration:

    mov [rel k], rcx

    mov r10, [rel i]
    dec r10
    imul r10, [rel N]
    add r10, [rel k]
    dec r10

    imul r10, dword 8      ; вычисляем сдвиг
указателя

    mov rax, [rel matrix]
    add rax, r10

    mov r10, [rel k]
    dec r10
    imul r10, [rel N]
    add r10, [rel j]
    dec r10

    imul r10, dword 8      ; вычисляем сдвиг
указателя

    mov rcx, [rel matrix2]
    add rcx, r10

    mov r11, [rel rax]      ; умножаем значение
элемента строки первой матрицы

    imul r11, [rel rcx]      ; на значение
элемента столбца второй матрицы

    add r12, r11      ; суммируем так все
произведения

    mov rcx, [rel k]

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

loop ElemIteration

    mov r10, [rel i]

    dec r10

    imul r10, [rel N]

    add r10, [rel j]

    dec r10

    imul r10, dword 8      ; вычисляем сдвиг указателя

    mov rax, [rel answer_matrix]

    add rax, r10

    mov [rel rax], r12      ; записываем сумму
произведений элементов

                                ; i-ой строки первой
матрицы на j-й столбец второй

    mov rcx, [rel j]

    dec rcx

    cmp rcx, [rel zero]

    jnz ColumnIteration

    mov rcx, [rel i]

    dec rcx

    cmp rcx, [rel zero]

    jnz RowsIteration

    ; N * num_of_column + num_of_row

    ; N * num_of_row + num_of_column

    xor rax, rax

    pop rbx

    ret

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```
_WrongInput:

    mov rdi, wrongInput
    call _printf

    pop rbp

    jmp _main

ExitProgram:

    mov     rax, 0           ; normal, no error, return value
    pop     rbp
    ret
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

