ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет КНТ

Кафедра ПИ

Лабораторная работа №5

по курсу: «Архитектура компьютера»

по теме: «Ассемблер. Подпрограммы. Способы адресации данных при обработке и вычислениях матрицы.»

Выполнил:

ст. гр. ПИ-19а

Саевский Олег

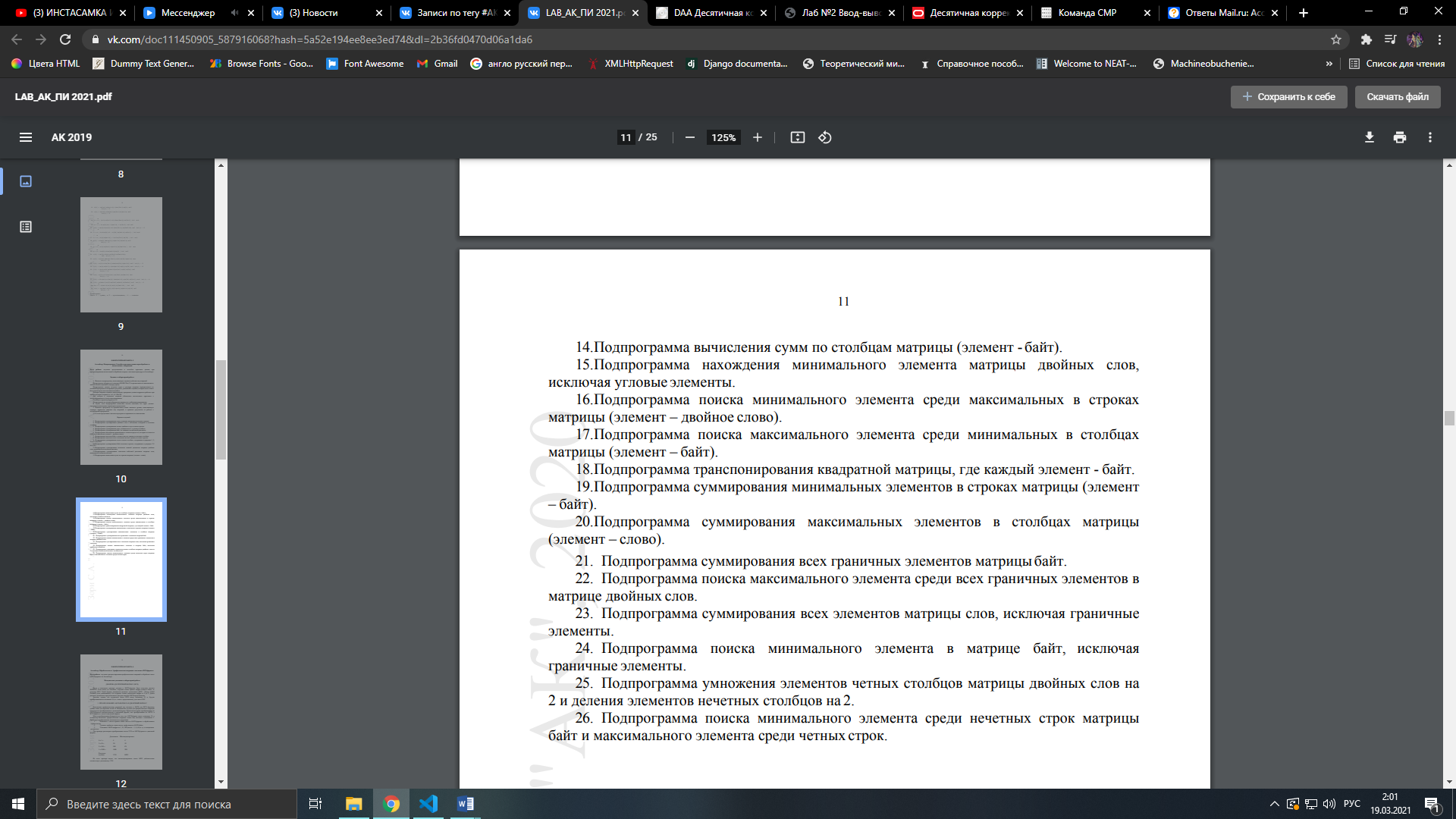
Проверил:

Павлюк Елена Николаевна

Московченко Анастасия Валерьевна

ДОНЕЦК – 2021

Задание



Код программы

format PE console

entry start

include 'win32ax.inc'

section '.data' data readable writable

    arr dd 0 dup (5)

    buff dd 2

    num dd 0

section '.code' code readable writable

    start:

        .col:

        cinvoke printf, <'%s', 0xA>, 'Input count colums of matrix: '

                call int\_input

        cmp[buff], 0

        jle .col

        mov edi, [buff]; eax - размер матрицы

        mov ebp, edi

        mov ebx, 1

        .row:

        invoke printf, <'%s', 0xA>, 'Input count rows of matrix: '

            call int\_input

        cmp [buff], 0

        jle .row

        cinvoke printf, <'%s', 0xA>, 'Input matrix: '

        mov esi, [buff]

        mov ecx, esi

        .for1:

            mov edi, ebp

            .for2:

                push ecx

                call int\_input

                mov [arr+4\*ebx], edx

                sub edi,1

                inc ebx

                pop ecx

                cmp edi, 0

            jnle .for2

        loop .for1; ebx < eax ebx < eax

        cinvoke printf, <'%s', 0xA>, 'Your matrix:'

        mov ebx, 1

        mov ecx, esi

        mov esi, [arr + 4\*ebx]

        mov eax, 2147483647

         .for3:

            push eax

            push ecx

            cinvoke printf, <'%s', 0xA>, ' '

            pop ecx

            pop eax

            mov edi, ebp

            mov esi, [arr + 4\*ebx]

            .for4:

                push eax

                push ecx

                cinvoke printf, <'%d '>, [arr + 4\*ebx]

                cmp esi, [arr + 4\*ebx]

                jg .cont

                mov esi, [arr + 4\*ebx]

                jmp .cont

                .cont:

                sub edi,1

                inc ebx

                pop ecx

                pop eax

                cmp edi, 0

            jnle .for4

            cmp eax, esi

            jle .cont2

            mov eax, esi

            jmp .cont2

            .cont2:

        loop .for3; ebx < eax ebx < eax

        push eax

        cinvoke printf, <'%s', 0xA>, ' '

        cinvoke printf, <'%s'>, ' Min Value = '

        pop eax

        cinvoke printf, <'%d', 0xA>, eax

        cinvoke system, 'pause'

        cinvoke ExitProcess, 0

int\_input:

    .input:

        cinvoke scanf, '%d', buff

        cmp eax, 1

        jne .error

        mov edx, [buff]

        ret

    .error:

        cinvoke scanf, '%c', buff

        jmp .input

section '.idata' import data readable

    library kernel32, 'kernel32.dll',\

            msvcrt, 'msvcrt.dll'

    import kernel32,\

           ExitProcess, 'ExitProcess'

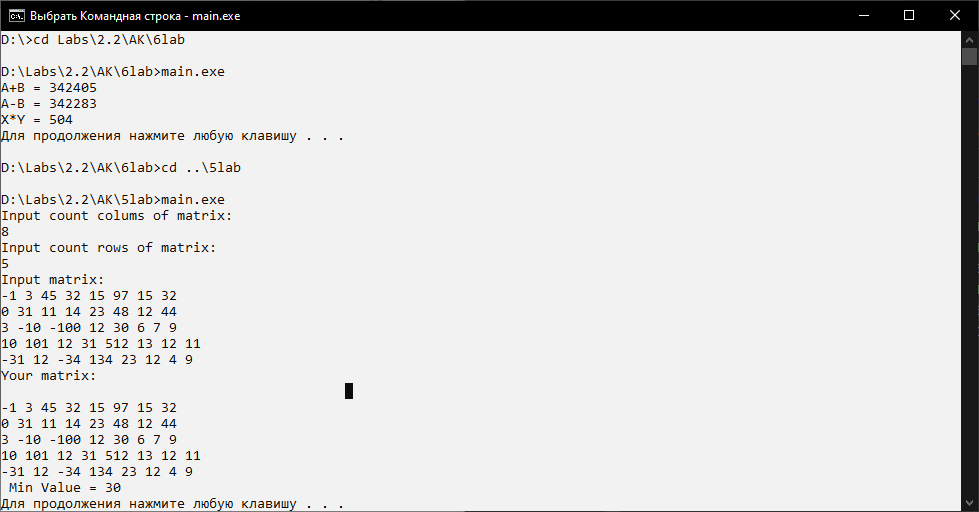
    import msvcrt,\

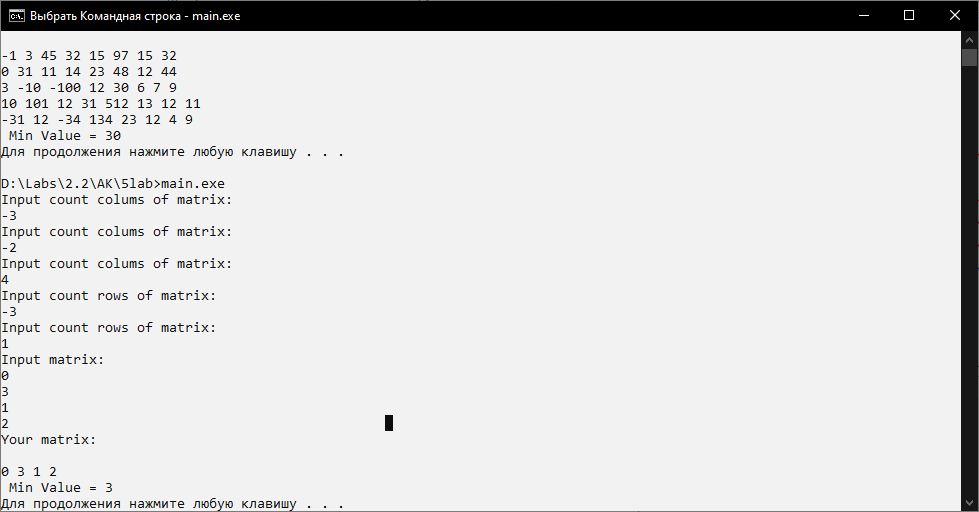
           printf, 'printf',\

           scanf, 'scanf',\

           system, 'system'

Пример работы программы





Код на языке Python

matrix = [[-1, 3, 45, 32, 15, 97, 15, 32], [0, 31, 11, 14, 23, 48, 12, 44], [3, -10, -100, 12, 30, 6, 7, 9], [10, 101, 12, 31, 512, 13, 12, 11], [-31, 12, -34, 134, 23, 12, 4, 9]]

min = 3223532

for item in matrix:

    max = item[0]

    for i in item:

        if max <= i:

            max = i

    if min >= max:

        min = max

print("Min value:", min)

Результат

