ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет КНТ

Кафедра ПИ

Лабораторная работа №6

по курсу: «Архитектура компьютера»

по теме: «Ассемблер. Арифметические операции в BCD - формате.»

Выполнил:

ст. гр. ПИ-19а

Саевский Олег

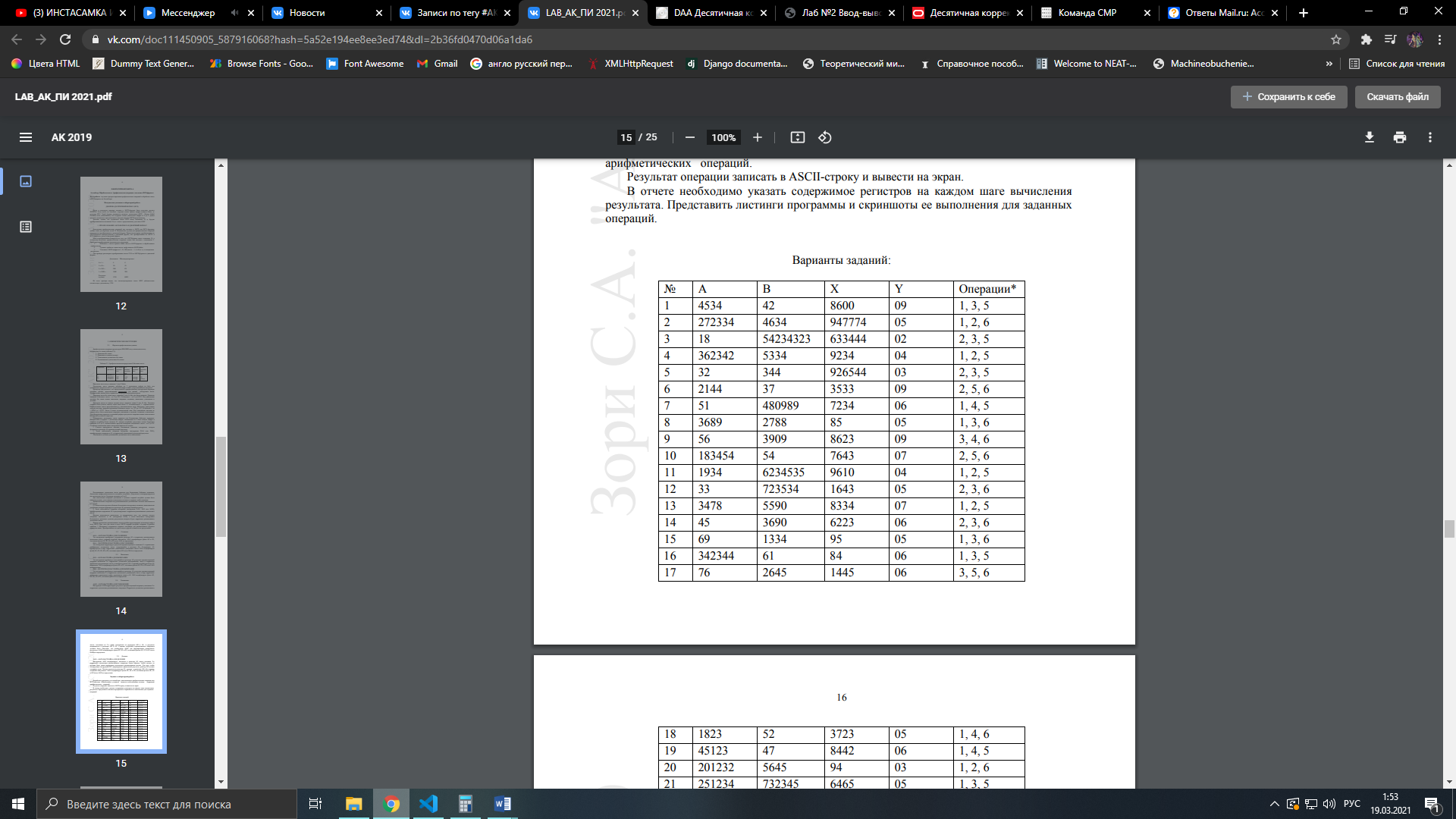
Проверил:

Павлюк Елена Николаевна

Московченко Анастасия Валерьевна

ДОНЕЦК – 2021

Задание



Код программы

format PE console

entry start

include 'win32ax.inc'

section '.data' data readable writable

        a dd 342344h

        b dd 61h

        v1 rb 0

        v2 rb 0

        x db 8,4

        y db 6

        res1 dd 0

        res2 dd 0

        res3 rb 1

section '.code' code readable executable

start:

        ; cinvoke system, 'color f0'

        xor eax, eax

        xor ebx, ebx

        mov ecx, 3

        for1:

                mov al, byte ptr a+ebx

                adc al, byte ptr b+ebx  ; sbb - вычитание с учётом займа

                daa

                mov byte ptr res1+ebx, al

                inc ebx

                loop for1

        cinvoke printf, <"A+B = %x", 0xA>, [res1]

        xor eax, eax

        xor ebx, ebx

        mov ecx, 3

        for2:

                mov al, byte ptr a+ebx

                sbb al, byte ptr b+ebx  ; sbb - вычитание с учётом займа

                das     ; Decimal Adjust After Subtraction

                mov byte ptr res2+ebx, al

                inc ebx

                loop for2

        cinvoke printf, <"A-B = %x", 0xA>, [res2]

        cinvoke printf, 'X\*Y = '

        xor eax, eax

        xor ebx, ebx

        mov al, [x+1]

        mov bl, [y]

        mul [y]

        aam

        mov [res3], al

        mov [res3 + 1], ah

        mov al, [x]

        mul [y]

        aam

        adc al, byte ptr v1

        daa

        cmp al, 10

        jl .Ok

        mov [res3 + 1], 0

        mov [res3 + 2], ah

        add [res3 + 2], 1

        mov ecx, 3

        jmp .Ok2

        .Ok:

        adc [res3 + 1], ah

        mov ecx, 2

        .Ok2:

        xor eax, eax

        xor ebx, ebx

        .for3:

                xor edx, edx

                mov dl, [res3+ecx-1]

                push ecx

                cinvoke printf, <"%x">, edx

                pop ecx

                inc ebx

                loop .for3

        cinvoke printf, '%c', 0xA

        cinvoke system, 'pause'

        cinvoke ExitProcess, 0

section '.idata' import data readable

        library kernel32, 'kernel32.dll',\

                msvcrt, 'msvcrt.dll'

        import kernel32,\

                ExitProcess, 'ExitProcess'

        import msvcrt,\

                printf, 'printf',\

                system, 'system'

Пример работы программы

