**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"**

Лабораторная работа № 4

Тема: «Ассемблер. Способы адресации данных. Программирование циклических процессов и работа с массивами»

Проверил: Выполнил:

асс. каф. ПИ ст. гр. ПИ-18б

Ищенко А.П. Моргунов А. Г.

\_\_\_\_.\_\_\_\_.2020г. \_\_\_\_.\_\_\_\_.2020г.

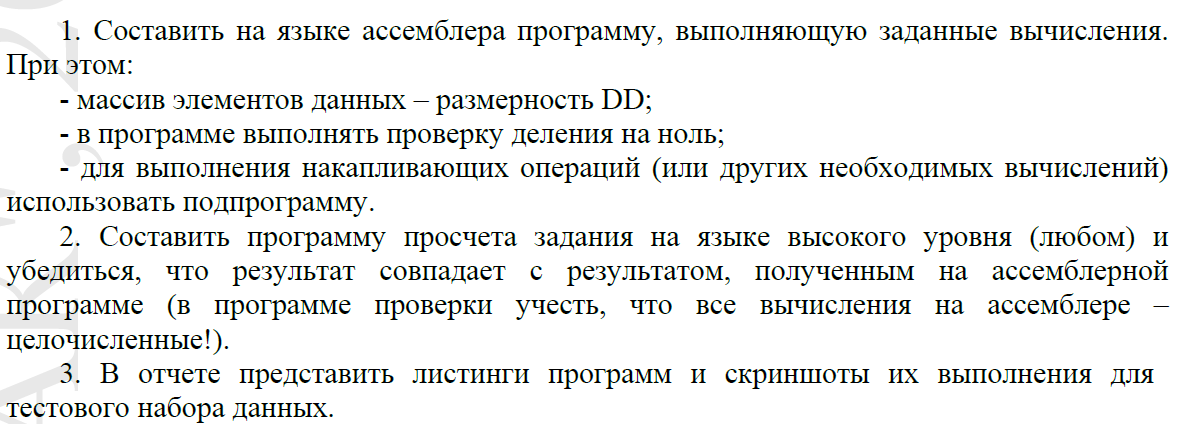
асс. каф. ПИ

Павлюк Е.Н.

\_\_\_\_.\_\_\_\_.2020г.

Донецк – 2019

Задание:





Код на ассемблере

.386

.model flat, stdcall

option casemap : none

.data

DivZ dd 0

N dd 5

k dd 1

A dd 8,65,19,11,12,16

M dd 1,4,1,2,1,3,1

Z dd 5 dup(?)

.code

;функция k\*M[k]

imulEBX PROC

imul ebx,k ;ebx->k\*M[k]

cdq ;дублируем знак eax в edx

ret

imulEBX ENDP

main PROC

mov ecx, N

forN:

mov esi, k ;k-индекс элемента

mov eax, A[esi\*4];eax->A[k]

doneg: ;

neg eax ;eax=|A[k]|

js doneg ;

mov ebx, A[ecx\*4] ;ebx->A[N+1-k]

cmp ebx, 0 ;A[N+1-k] ?= 0

je DivByZero ;A[N+1-k] == 0 -> DivByZero

cdq ;дублируем знак eax в edx

idiv ebx ;eax-> |A[k]|/A[N+1-k]

mov ebx, M[ecx\*4+4] ;ebx->M[N+2-k]

cmp ebx, 0 ;M[N+2-k] ?= 0

je ZDiv ;M[N+2-k] == 0 -> ZDiv

cdq ;дублируем знак eax в edx

idiv ebx ;eax-> |A[k]|/A[N+1-k]/M[N+2-k]

mov ebx,M[esi\*4] ;ebx->M[k]

call imulEBX

sub ebx,eax ;ebx->k\*M[k] - |A[k]|/A[N+1-k]/M[N+2-k]

mov Z[esi\*4],ebx ;z[k] = k\*M[k] - |A[k]|/A[N+1-k]/M[N+2-k]

push ebx ;z[k] в стек

inc k ;k+1

loop forN ;цикл

pop edi ;z[5]

pop edx ;z[4]

pop ecx ;z[3]

pop ebx ;z[2]

pop eax ;z[1]

ret

ZDiv: ;деление на 0

mov DivZ, 1 ;DivZ -> 1 флаг деления на 0

ret

main ENDP

END main

Код на С++

#include <iostream>

#include <cstdlib>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

int Z[6];

int M[7] = {1,4, 1, 2, 1, 3, 1};

int A[6] = {8,65,19,11,12,16};

int N = 5;

for (int k = 1; k < 6; k++) {

if (A[N + 1 - k] == 0 || M[N + 2 - k] == 0) {

cout << "M[" << k << "] = " << M[k] << endl << "A[" << N + 1 - k << "] = " << A[N + 1 - k] << endl << "M[" << N + 2 - k << "] = " << M[N + 2 - k] << endl;

cout << "Деление на 0" << endl << endl;

}

else {

Z[k] = k \* M[k] - abs(A[k]) / A[N + 1 - k] / M[N + 2 - k];

cout << "M[" << k << "] = " << M[k] << endl << "A[" << N + 1 - k << "] = " << A[N + 1 - k] << endl << "M[" << N + 2 - k << "] = " << M[N + 2 - k] << endl;

cout << "Z[" << k << "] = " << k << " \* " << M[k] << " - abs(" << A[k] << ") / " << A[N + 1 - k] << " / " << M[N + 2 - k] << " = " << Z[k] << endl << endl;

} }

system("pause");}

Вывод на ассемблере соответствует выводу на С++

