ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет КНТ

Кафедра ПИ

Лабораторная работа №4

по курсу: «Архитектура компьютера»

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

Выполнил:

ст. гр. ПИ-19а

Саевский Олег

Проверил:

Павлюк Елена Николаевна

Московченко Анастасия Валерьевна

ДОНЕЦК – 2021

Вариант 16

Задание



Исходный код программы

#include <iostream>

#define N 6

// k = 1, N = 6. A, B

int getValue()

{

while (true) // цикл продолжается до тех пор, пока пользователь не введёт корректное значение

{

int a;

std::cin >> a;

// Проверка на предыдущее извлечение

if (std::cin.fail()) // если предыдущее извлечение оказалось неудачным

{

std::cin.clear(); // то возвращаем cin в 'обычный' режим работы

std::cin.ignore(32767, '\n'); // и удаляем значения предыдущего ввода из входного буфера

std::cout << "К сожалению, ввод неверен. Пожалуйста, попробуйте еще раз.\n";

}

else

{

std::cin.ignore(32767, '\n'); // удаляем лишние значения

return a;

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

system("color F0");

int A[13],

B[8], S, Y = 0;

int k = 1, Nul;

for (int i = 0; i < 13; i++)

{

std::cout << "Введите число A[" << i << "] = ";

A[i] = getValue();

}

for (int i = 0; i < 8; i++)

{

std::cout << "Введите число B[" << i << "] = ";

B[i] = getValue();

}

std::cout << "A: ";

for (int i = 0; i < 13; i++) {

std::cout << A[i] << " ";

}

std::cout << "\n" << "B: ";

for (int i = 0; i < 8; i++) {

std::cout << B[i] << " ";

}

std::cout << "\n";

\_asm

{

jz Start

Start:

mov esi, k

mov ecx, k

imul ecx, k; k\*\* 2

imul ecx, [A + 4\*esi]; k\*\* 2 \* A[k]

imul ecx, [A + 4\*esi]; k\*\* 2 \* A[k] \* \*2

mov edi, -4

imul edi, esi

mov ebx, [A + 4 \* N + edi + 4]

sub ebx, [B + 4 \* esi + 4]

cmp ebx,0

je Zero

mov eax, A[4 \* 2 \* esi]

cdq; подготовка к делению

idiv ebx;

sub ecx, eax

add Y, ecx

add esi, 1

mov k, esi

cmp esi, N

jle Start

jmp OK

Zero:

mov Nul,1

jmp OK

OK:

}

if (Nul == 1)

{

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

}

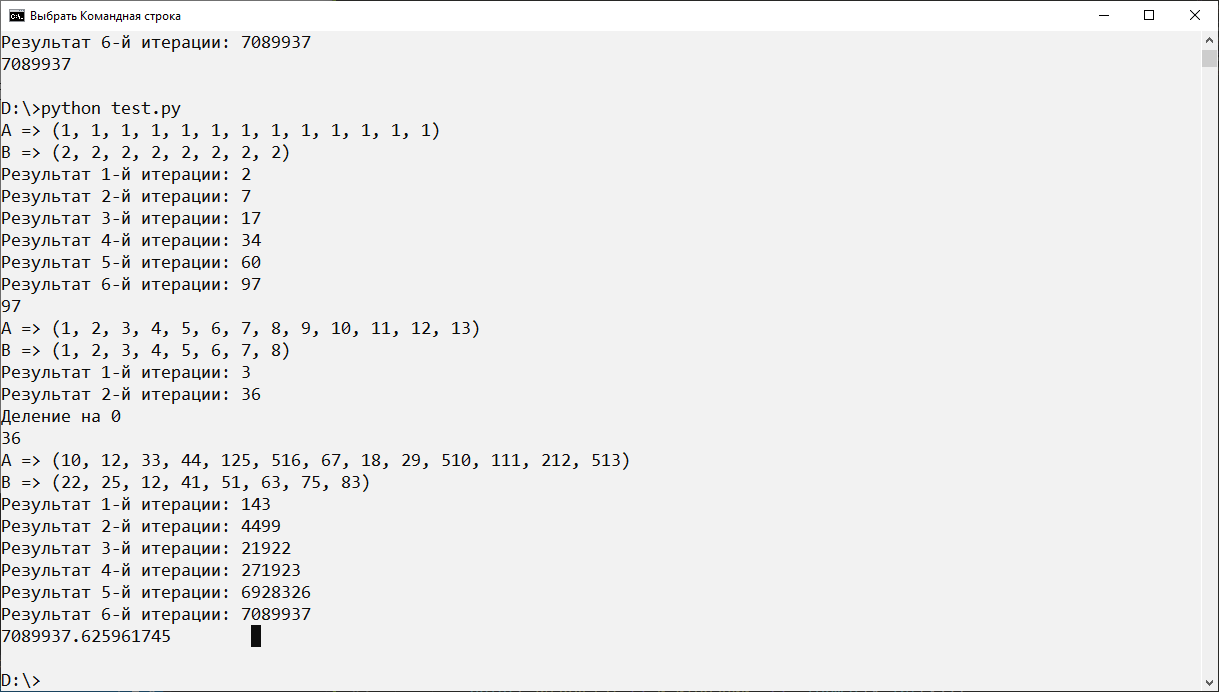
else

std::cout << "Y = " << Y << std::endl;

return 0;

}

Расчет результатов в Python



Код программы:

import os

os.system('color f0')

A = (1,1,1,1,1,1,1,1,1,1,1,1,1)

B = (2,2,2,2,2,2,2,2)

print("A => {}".format(A))

print("B => {}".format(B))

Y = 0

for k in range(1,7):

try:

Y = Y + k\*\*2 \* A[k]\*\*2 - A[2\*k] / (A[6 + 1 - k] - B[k + 1])

print("Результат {}-й итерации: {}".format(k,int(Y)))

except Exception as e:

print("Деление на 0")

break

print(int(Y))

A = (1,2,3,4,5,6,7,8,9,10,11,12,13)

B = (1,2,3,4,5,6,7,8)

print("A => {}".format(A))

print("B => {}".format(B))

Y = 0

for k in range(1,7):

try:

Y = Y + k\*\*2 \* A[k]\*\*2 - A[2\*k] / (A[6 + 1 - k] - B[k + 1])

print("Результат {}-й итерации: {}".format(k,int(Y)))

except Exception as e:

print("Деление на 0")

break

print(int(Y))

A = (10,12,33,44,125,516,67,18,29,510,111,212,513)

B = (22,25,12,41,51,63,75,83)

print("A => {}".format(A))

print("B => {}".format(B))

Y = 0

for k in range(1,7):

try:

Y = Y + k\*\*2 \* A[k]\*\*2 - A[2\*k] / (A[6 + 1 - k] - B[k + 1])

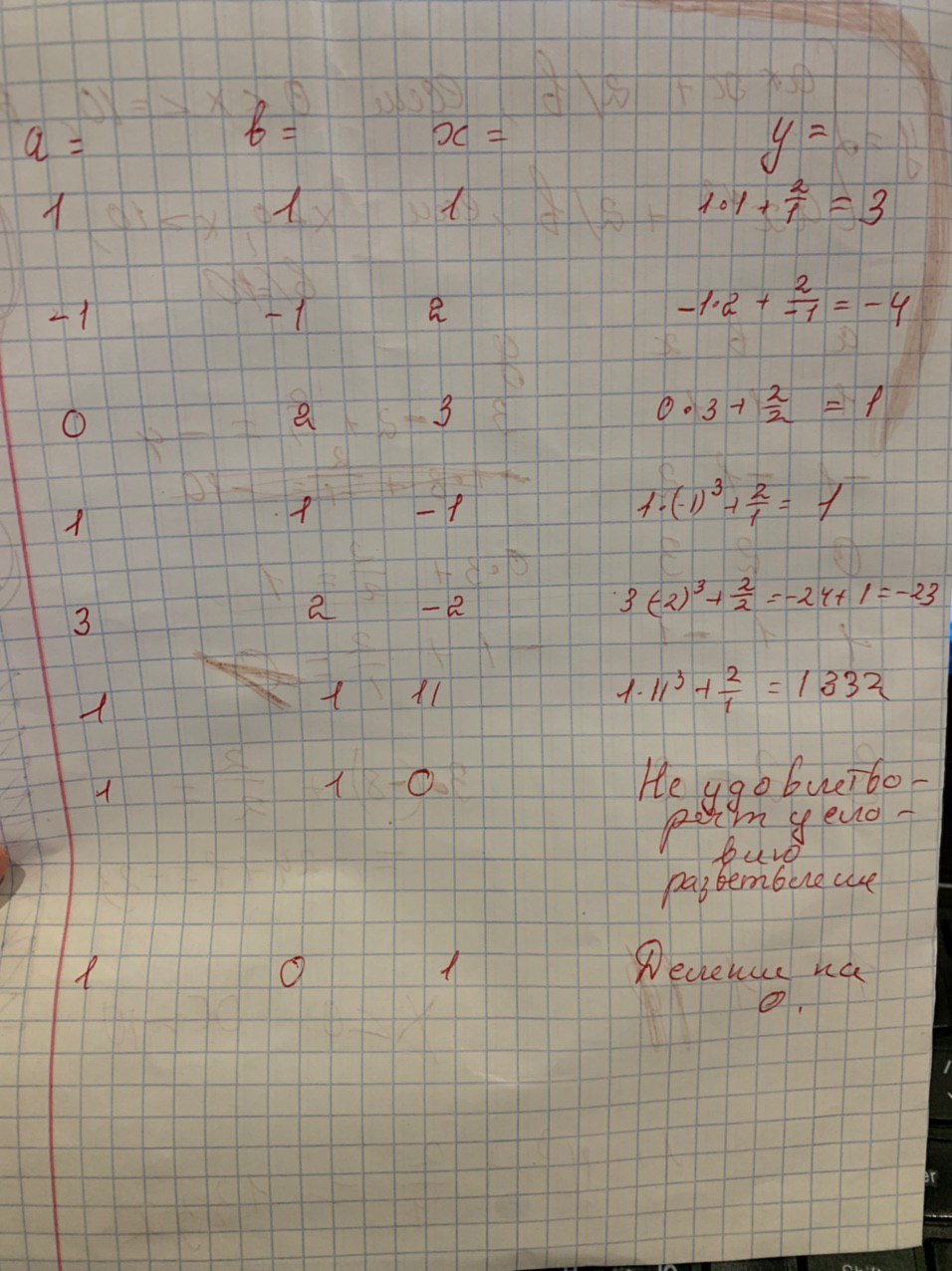
print("Результат {}-й итерации: {}".format(k,int(Y)))

except Exception as e:

print("Деление на 0")

break

print(Y)

Начало формыКонец формы

Скриншоты

