Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королева»

Институт информатики, математики и электроники

Факультет информатики

Кафедра технической кибернетики

Отчет по лабораторной работе №3

Дисциплина: «Системное программирование и архитектура ЭВМ»

Выполнил: Абрамов Д.О.

Группа: 6208-010302D

Самара, 2021

**Задание на лабораторную работу**

1) В программе необходимо реализовать функцию вычисления заданного условного целочисленного выражения, используя команды сравнения, условного и безусловного переходов на встроенном ассемблере.

2) Результат X – целочисленный, возвращается из функции регистре eax.

3) Значения переменных передаются в качестве параметров функции.

4) В программе реализовать вывод результата на экран

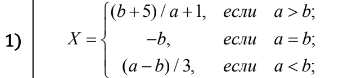
(не в функции вычисляющей выражение).

5) Все параметры функции 32 битные числа.

6) Проверку деления на 0 реализовать на встроенном ассемблере.

7) При необходимости организовать проверку переполнения (использовать результаты ЛР2)

8) В качестве комментария к каждой строке необходимо указать, какой промежуточный результат, в каком регистре формируется.



**Код программы**

#include <iostream>

#include <windows.h>

#include <tchar.h>

using namespace std;

/\*

(b + 5)/a + 2, a>b

x = -b, a=b

(a-b)/3, a<b

\*/

int main()

{

int error;

int a, b, res, res\_asm;

cout << "Please, enter a: ";

cin >> a;

cout << "Please, enter b: ";

cin >> b;

if (a > b && a == 0) {

cout << "Do not try to divide by zero\n";

system("pause");

return -1;

}

if (a > b) {

res = (b + 5) / a + 2;

}

else if (a == b) {

res = -b;

}

else {

res = (a - b) / 3;

}

\_\_asm {

mov eax, a; eax = a

mov ebx, b; ebx = b

cmp eax, ebx; сравнение a и b

jg l\_bigger; переход если a > b

jl l\_smaller; переход если a < b

neg ebx; ebx = -b

mov res\_asm, ebx; res\_asm = -b

jmp EndOfCode; переход на конец программы

l\_bigger :

mov eax, b; eax = b

mov ebx, a; ebx = a

jo error\_of

add eax, 5; eax = b + 5

cmp ebx, 0; сравнение a и 0

cdq;

je error\_zf; ошибка деление на ноль

idiv ebx; eax = (b + 5) / a

jo error\_of

add eax, 2; eax = (b + 5) / a + 2

mov res\_asm, eax; res\_asm = (b + 5) / a + 2

jmp EndOfCode; переход на конец программы

l\_smaller :

sub eax, ebx; eax = a - b

mov ebx, 3; ebx = 3

cdq

je error\_zf; ошибка деление на ноль

idiv ebx; eax = (a - b) / 3

mov res\_asm, eax; res\_asm = (a - b) / 3

jmp EndOfCode

error\_zf :

mov error, 1

jmp EndOfCode

error\_of :

mov error, 2

jmp EndOfCode

EndOfCode :

}

if (error == 1) { //errror\_zf

cout << "Do not try to divide by zero\n";

system("pause");

return 0;

}

if (error == 2) { //error\_of

cout << "Too big quotient\n";

system("pause");

return 0;

}

cout << "res c++ : " << res << endl;

cout << "res asm : " << res\_asm << endl;

system("pause");

return 0;

}