**Практическая работа № 8**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил | Студент гр. МФ-11 | Лазаренко А. В. | 12.04.2021 |
| Проверил | Ст. преподаватель | Меняйлов Е. С. |  |

**Тема работы:**

«Организация условных переходов»

**Цель работы:**

Вычислить заданное целочисленное выражение для данных в формате INTEGER (байтовый инт) переменные a, b, а RESULT в формате LONGINT (4-байтовый инт), используя арифметические операции ADD, ADC, INC, SUB, SBB, DEC, NEG, MUL, IMUL, DIV, IDIV, CBW, CWD и, если нужно, логические операции SAL, SHL, SAR, SHR, а также команды сравнения, условного и безусловного переходов.

**Задание:**

1. Проанализировать арифметическое выражение, определить порядок

выполнения операций, типы получающихся значений, там, где возможно,

упростить.

2. Написать на языке ассемблера процедуру, вычисляющее заданное значение

по входным переменным.

3. Написать программу на C++/Assembler, которая вводит значения входных

параметров и осуществляет проверку на аномалии при необходимости выдает

предупреждающее сообщения.

4. Протестировать программу на корректных и аномальных входных данных.

**Вариант 11:**

**Номер 1**

#include <iostream>

using namespace std;

int main(void)

{

short int ax, bx;

cin >> ax >> bx;

if ((ax > 127) || (ax < -128) || (bx > 127) || (bx < -128))

{

cout << "\nYou have entered wrong variable values.\n";

return 1;

}

char a = char(ax);

char b = char(bx);

long int result, result\_second;

result = result\_second = 0;

\_\_asm

{

mov al, a;

cmp al, b;

je \_equal

jg \_greater

jl \_lower

\_equal :

{

mov eax, -2;

jmp \_exitt;

}

\_greater:

{

sal al, 1;

add al, b;

cbw;

movsx eax, ax;

cbw;

jmp \_exitt;

}

\_lower:

{

mov al, a;

sub al, 5;

cbw;

movsx ebx, ax;

mov al, b;

cbw;

mov cx, ax;

movsx ecx, cx;

mov eax, ebx;

cdq;

cmp ecx, 0;

je \_divzero;

idiv ecx;

mov result\_second, edx;

jmp \_exitt;

}

\_divzero:

{

mov result\_second, SHRT\_MIN;

mov ax, 4C00h;

}

\_exitt:

{

mov result, eax;

}

}

if (result\_second == SHRT\_MIN)

printf("\nThe denominator is equal to zero.\n");

else

printf("\nWhole part: %ld; Remainder: %ld\n", result, result\_second);

return 0;

}

**Результаты выполнения программы:**

1)

2)

3)



4)



5) 

**Задание\***

#include <iostream>

using namespace std;

int main(void)

{

long double a = -2.0f;

long double b = 1.5f;

/\* Блок обязательных для вычисления переменных \*/

float x = 2.0f;

float y = -2.0f;

float z = 5.0f;

long double result = 0.0f;

\_\_asm

{

fld b; // помещаю в регистр st0 значение b

fld a; // помещаю в регистр st0 значение a

fcomi st(0), st(1); // сравниваю значения регистров st0 и st1 (a и b)

je \_equal; // если значения регистров равны

ja \_greater; // если значение st0 > st1

jb \_lower; // если значение st0 < st1

\_equal:

fld y; // помещаю в регистр st0 значение y

jmp \_exitt; // перехожу по метке \_exitt

\_lower:

fld z; // помещаю в регистр st0 значение z

fld a; // помещаю в регистр st0 значение a

fsub st(0), st(1); вычитаю регистры st0 и st1; результат в st0

fdiv b; // делю значение в st0 на значение b

jmp \_exitt; // перехожу по метке \_exitt

\_greater:

fmul x; // умножаю значение в st0 на значение x

fadd b; // складываю st0 и b; результат в st0

jmp \_exitt; // перехожу по метке \_exitt

\_exitt:

fst result; // помещаю значение регистра st0 в переменную result

}

if (result == SHRT\_MIN)

printf ("\nThe denominator is equal to zero.\n");

else

printf("%.3lf.\n", result);

return 0;

}

**Результаты выполнения программы:**

1) 

2) 

3) 

**Вывод:**

Я сумел вычислить заданное целочисленное выражение для данных в формате INTEGER (байтовый инт) переменные a, b, а RESULT в формате LONGINT (4-байтовый инт), используя арифметические операции ADD, SUB, IMUL, IDIV, CBW, CWD, CDQ и логические операции SAL, SHL, а также команды сравнения и условного перехода.