МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,

СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

Факультет Информационных технологий и программной инженерии

Кафедра Программной инженерии и вычислительной техники

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине:

**«Машинно-зависимые языки программирования»**

тема: Вычисление целочисленных арифметических операций

Выполнил студент(ка):

Мананков Антон Павлович

*(Ф.И.О., № группы) (подпись)*

Дата выполнения:

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

Проверил:

Савельев Игорь Леонидович

*(Ф.И.О.) (подпись)*

Санкт-Петербург

2025

**1. Задание**

Вычислить заданное целочисленное выражение(Рисунок 1) для исходных данных в знаковых и беззнаковых форматах длиной 8 и 16 бит: **signed char**, **unsigned short int.**

Рисунок 1 - Арифмитическое выражение

**2. Текст программы**

**2.1. Модуль main.c**

#include <stdio.h>

#include <stdint.h>

int8\_t a, b, c;

int16\_t num1, num2, res1;

uint16\_t a\_2, b\_2, c\_2;

int64\_t num3, num4, res2;

extern void func8bit();

extern void func16bit();

//Функция для вычисления на си 8 бит

void c\_calculate\_8bit(int8\_t a, int8\_t b, int8\_t c) {

int16\_t num, den, res;

printf("debug: %d %d %d\n", a, b, c);

num = (b\*7+64/a);

den = (31-c\*b/2);

if (den != 0) {

res = num / den;

} else {

printf("Divide by 0");

}

printf("Numerator is %d\n", num);

printf("Denominator is %d\n", den);

printf("Result is %d\n\n", res);

}

//Функция для вычисления на си 16 бит

void c\_calculate\_16bit(uint16\_t a, uint16\_t b, uint16\_t c) {

int64\_t num, den, res;

printf("debug: %d %d %d\n", a, b, c);

num = ((int64\_t)b\*7+64/(int64\_t)a);

den = (31-(int64\_t)c\*(int64\_t)b/2);

if (den != 0) {

res = num / den;

} else {

printf("Divide by 0");

}

printf("Numerator is %lld\n", num);

printf("Denominator is %lld\n", den);

printf("Result is %lld\n\n", res);

}

int main() {

printf("\n\n(b\*7+64/a)/(31-c\*b/2)\n\n");

printf("Enter the values for 8-bit (signed char from -128 to 127):\n");

printf("a = ");

scanf("%hhd", &a); //hh - signed char, d - чтобы вывело как число

printf("b = ");

scanf("%hhd", &b);

printf("c = ");

scanf("%hhd", &c);

// Вычисления на Си (8-бит)

printf("\nCalculation on C 8-bit (signed char)\n");

c\_calculate\_8bit(a, b, c);

// Вычисления на ASM (8-бит)

func8bit();

printf("\nCalculation on ASM 8-bit (signed char)\n");

printf("num=%d\nden=%d\nres=%d\n\n", num1, num2, res1);

printf("\nEnter the values for 16-bit (unsigned short from 0 to 65535):\n");

printf("a = ");

scanf("%hu", &a\_2); //hu - так википедия написала (h - char, u - unsigned)

printf("b = ");

scanf("%hu", &b\_2);

printf("c = ");

scanf("%hu", &c\_2);

// Вычисления на Си (16-бит)

printf("\nCalculation on C 16-bit (unsigned short)\n");

c\_calculate\_16bit(a\_2, b\_2, c\_2);

// Вычисления на ASM (16-бит)

func16bit();

printf("\nCalculation on ASM 16-bit (unsigned int)\n");

printf("num= %lld\nden= %lld\nres= %lld\n\n", num3, num4, res2);

return 0;

}

**2.2. Модуль asm8bit.asm**

section .text

extern b, a, c, num1, num2, res1

global func8bit

func8bit:

; (1)

MOV AL, byte [b]

MOV BL, 7

IMUL BL ; al\*bl->ax

MOV CX, AX

; (2)

MOV AX, 64

MOV BL, byte [a]

IDIV BL

; (3)

MOVSX BX, AL

ADD BX, CX

MOV [num1], BX

; (4)

MOV AL, byte [b]

MOV BL, byte [c]

IMUL BL

CWD

; (5)

MOV BX, 2

IDIV BX

; (6)

MOV BX, 31

SUB BX, AX

MOV [num2], BX

; (7)

MOV AX, [num1]

CWD

IDIV word [num2]

MOV [res1], AX

ret

**2.3. Модуль asm16bit.asm**

section .text

extern b\_2, a\_2, c\_2, num3, num4, res2

global func16bit

func16bit:

; (1)

MOVZX EAX, word [b\_2]

MOV EBX, 7

MUL EBX ; eax\*ebx->edx:eax

MOV RCX, RAX

; (2)

MOV AX, 64

MOV BX, word [a\_2]

CWD

DIV BX

MOVZX RAX, AX

; (3)

ADD RAX, RCX

MOV [num3], RAX

; (4)

MOVZX EAX, word [b\_2]

MOVZX EBX, word [c\_2]

MUL EBX

; (5)

MOV EBX, 2

DIV EBX

; (6)

MOV RBX, 31

SUB RBX, RAX

MOV [num4], RBX

; (7)

MOV RAX, [num3]

MOV RBX, [num4]

CQO

IDIV RBX

MOV [res2], RAX

ret

**2.3. Модуль makefile**

all: lr1.exe

lr1.exe: main.c asm8bit.o asm16bit.o

gcc -m64 -o lr1.exe main.c asm8bit.o asm16bit.o -no-pie -z noexecstack

asm8bit.o: asm8bit.asm

nasm -f elf64 -F dwarf -g asm8bit.asm

asm16bit.o: asm16bit.asm

nasm -f elf64 -F dwarf -g asm16bit.asm

clean:

rm \*.exe

rm \*.o

**3. Сборка проекта**

sudo apt update && sudo apt install -y nasm make gcc git

git clone <https://github.com/IglFox/MZYAP>

cd LabWork\_1 && make

**4. Выполнение программы**

**4.1. Запуск программ**

./lr1.exe

**4.2. Входные данные**

Для 8 бит:

Enter the values for 8-bit (signed char from -128 to 127):

a = 1

b = 1

c = 1

Для 16 бит:

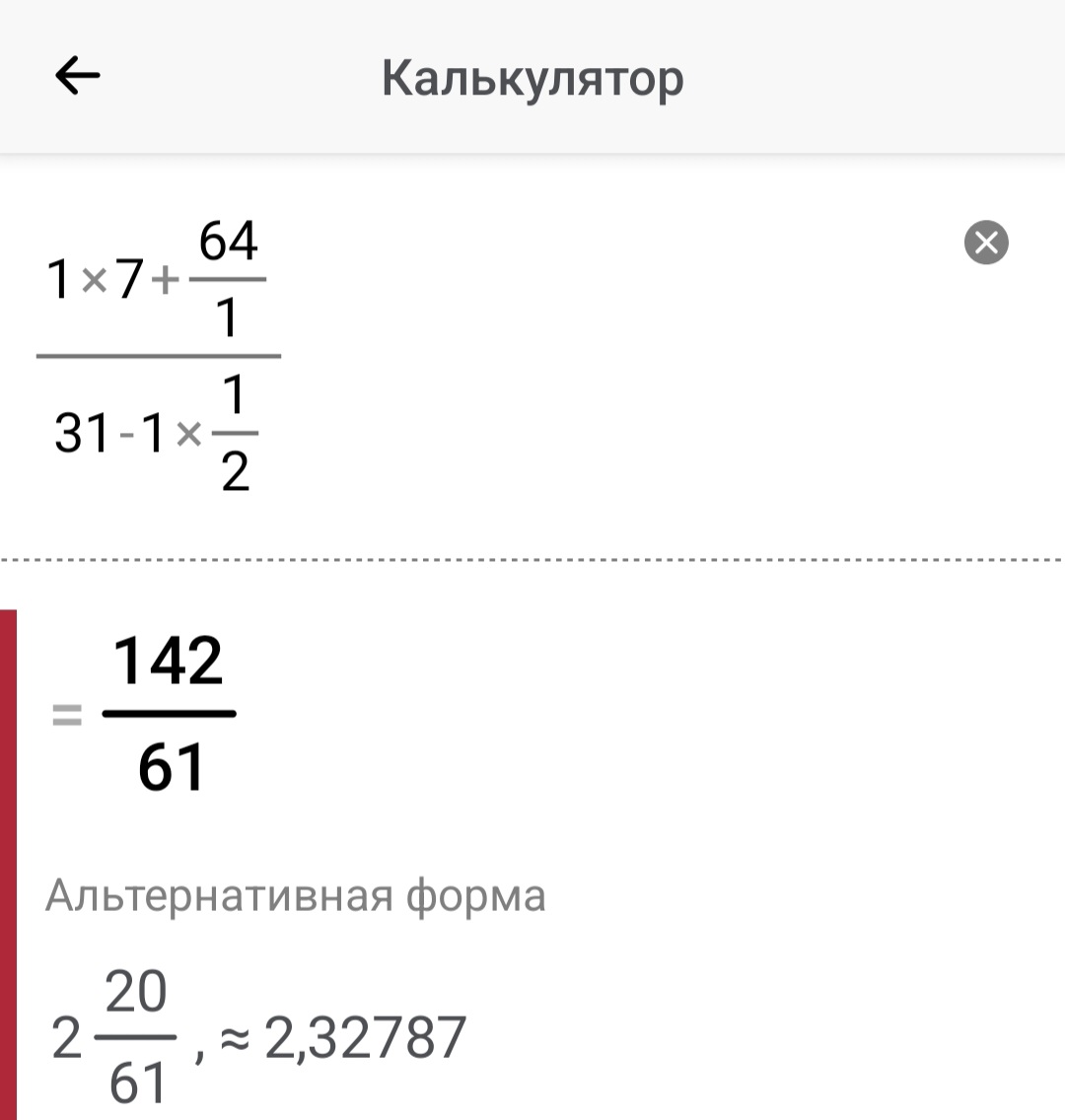
Enter the values for 16-bit (unsigned short from 0 to 65535):

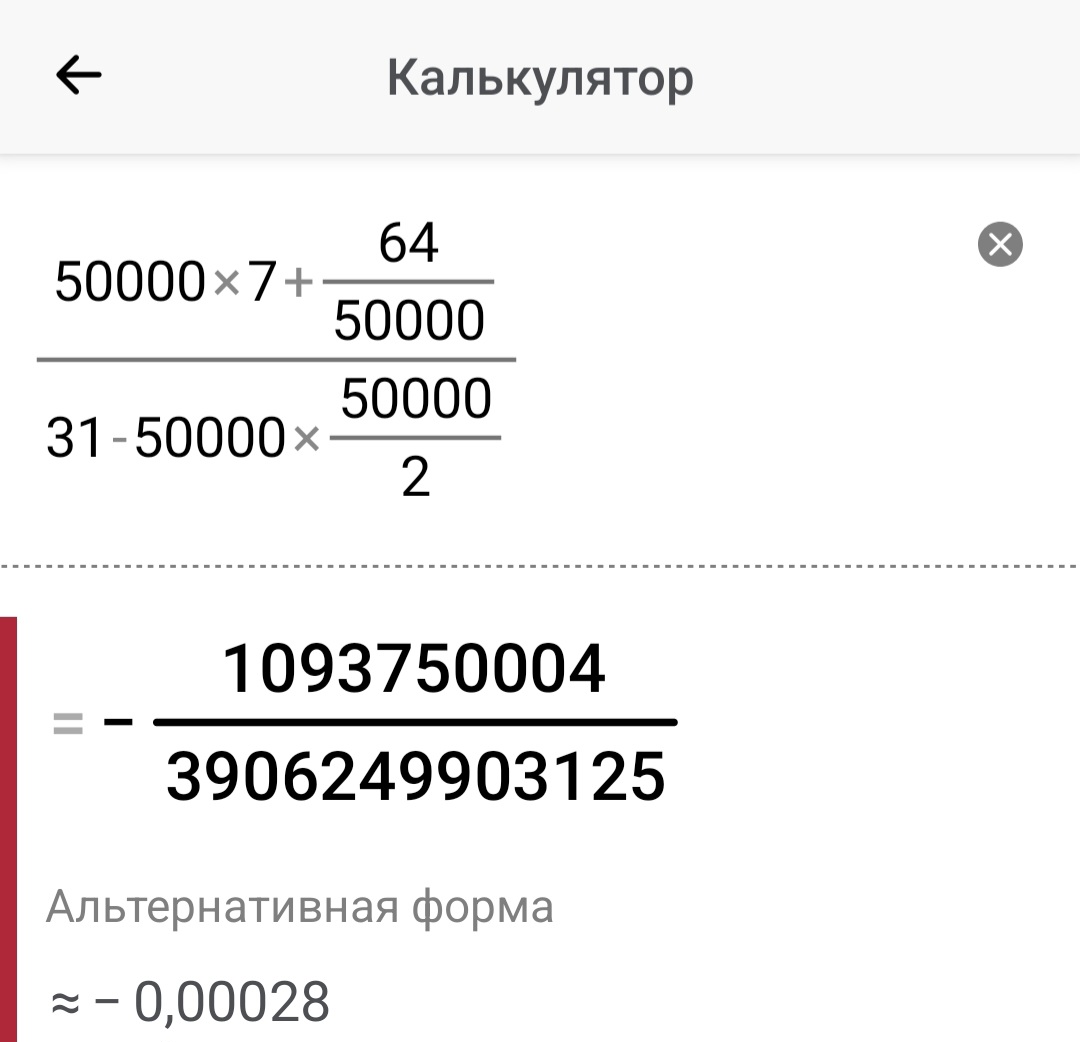
a = 65535

b = 65535

c = 65535

**4.3. Ожидаемый результат выполнения**

Рисунок 2 - Ожидаемый результат 8бит

Рисунок 3 - Ожидаемый результат 16бит

**4.4. Результат выполнения**

Для 8 бит:

Calculation on C 8-bit (signed char)

Numerator is 71

Denominator is 31

Result is 2

Calculation on ASM 8-bit (signed char)

num=71

den=31

res=2

Для 16 бит:

Calculation on C 16-bit (unsigned short)

Numerator is 458745

Denominator is -2147418081

Result is 0

Calculation on ASM 16-bit (unsigned int)

num= 458745

den= -2147418081

res= 0

**5. Вывод**

Результат выполнения программы соответствует ожидаемому результату. Работа выполнена в полном объеме.