Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Архитектура вычислительных систем

Отчёт по лабораторной работе №4-6

АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

С ЧИСЛАМИ С ПЛАВАЮЩЕЙ ТОЧКОЙ

Выполнил:

студент гр. 153503

Кончик Д. С.

Проверил:

Калиновская А.А.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc118909172)

[2 Задание 3](#_Toc118909173)

[3 Результат выполнения 8](#_Toc118909174)

[4 Вывод 9](#_Toc118909175)

**1 ЦЕЛЬ РАБОТЫ**

Изучить принципы представления чисел в формате с плавающей точкой в памяти компьютера; операции сложения, вычитания, умножения и деления чисел с плавающей точкой. Составить программу эмулятора АЛУ, реализующего соответствующие операции.

2 ЗАДАНИЕ

Написать программу эмулятора АЛУ, реализующего операции сложения, вычитания, умножения и деления с плавающей точкой над двумя введенными числами с возможностью пошагового выполнения алгоритмов.

Листинг 1 ­– Файл «main.cpp»

#include <iostream>

#include "Addition.h"

#include "Subtraction.h"

#include "Multiplication.h"

#include "Division.h"

using namespace std;

void Operation(void(\*operation)(float, float), float a, float b) {

try {

operation(a, b);

}

catch (string ex) {

cout << "Error: " << ex << "\n\n";

}

}

int main()

{

setlocale(LC\_ALL, "RU");

float a;

cout << "a = "; cin >> a;

float b;

cout << "b = "; cin >> b;

cout << "\n\n";

Operation(Addition, a, b);

cout << "\n\n";

Operation(Subtraction, a, b);

cout << "\n\n";

Operation(Multiplication, a, b);

cout << "\n\n";

Operation(Division, a, b);

return 0;

}

Листинг 2 ­– Файл «Addition.h»

#pragma once

#include <iostream>

#include <string>

#include "Helpers.h"

using namespace std;

void Addition(string a, string b) {

cout << "Сложение" << endl;

Print("a", a);

Print("b", b);

string ans = string(32, '0');

if (!IsZero(a) && !IsZero(b)) {

if (Exp(a) > Exp(b)) {

cout << "Порядок 'a' выше порядка 'b'. Меняем 'a' и 'b' местами.\n";

swap(a, b);

Print("a", a);

Print("b", b);

}

cout << "Включаем неявный старший разряд мантиссы.\n";

int SignA = Sign(a), SignB = Sign(b);

int ExpA = Exp(a), ExpB = Exp(b);

int MantA = Mant(a), MantB = Mant(b);

if (ExpA < ExpB) {

cout << "Порядки чисел не равны. Делаем выравнивание порядков.\n";

while (ExpA < ExpB) {

ExpA++;

MantA >>= 1;

}

if (ExpB < ExpA) {

throw string("Ошибка в выравивании порядков.");

}

}

cout << "a = " << (SignA == 1 ? "0" : "1") << " " << bitset<8>(ExpA + 127).to\_string() << " " << bitset<24>(MantA).to\_string() << endl;

cout << "b = " << (SignB == 1 ? "0" : "1") << " " << bitset<8>(ExpB + 127).to\_string() << " " << bitset<24>(MantB).to\_string() << endl;

if (MantA == 0) {

cout << "a ≈ 0.0\n";

ans = b;

}

else {

int M = MantA \* SignA + MantB \* SignB;

int E = ExpA;

int S = 1;

if (M != 0) {

if (M < 0) {

S = -1;

M = -M;

}

// Нормализация

while (M >= (1 << 24)) {

M >>= 1;

E++;

if (E >= 128) {

cout << endl;

throw string("Переполнение порядка");

}

}

while ((M & (1 << 23)) == 0) {

M <<= 1;

E--;

if (E <= -127) {

cout << endl;

throw string("Потеря значимости порядка");

}

}

cout << endl;

SetSign(ans, S);

SetExp(ans, E);

SetMant(ans, M);

}

}

}

else {

if (IsZero(a)) {

ans = b;

}

else if (IsZero(b)) {

ans = a;

}

}

Print("a + b", ans);

return;

}

void Addition(float a, float b) {

Addition(String(a), String(b));

}

Листинг 3 ­– Файл «Substraction.h»

#pragma once

#include <iostream>

#include <string>

#include "Helpers.h"

#include "Addition.h"

using namespace std;

static void Subtraction(string a, string b) {

cout << "Вычитание\n";

Print("a", a);

Print("b", b);

Addition(a, Negate(b));

}

static void Subtraction(float a, float b) {

Subtraction(String(a), String(b));

}

Листинг 4 ­– Файл «Multiplication.h»

#pragma once

#include <iostream>

#include <string>

#include "Helpers.h"

using namespace std;

void Multiplication(string a, string b) {

cout << "Умножение\n";

Print("a", a);

Print("b", b);

if (IsZero(a)) {

cout << "a \* b = 0\n";

return;

}

if (IsZero(b)) {

cout << "a \* b = 0\n";

return;

}

int SignA = Sign(a), SignB = Sign(b);

int ExpA = Exp(a), ExpB = Exp(b);

int MantA = Mant(a), MantB = Mant(b);

int S = SignA \* SignB;

int E = ExpA + ExpB;

long long M = ((long long)MantA \* MantB) >> 23;

if (E >= 128) {

throw string("Переполнение порядка");

}

if (E <= -127) {

throw string("Потеря значимости порядка");

}

if (M == 0) {

throw string("Мантисса произведения 0");

}

while (M >= (1 << 24)) {

M >>= 1;

E++;

if (E >= 128) {

cout << endl;

throw string("Переполнение порядка");

}

}

if ((M & (1 << 23)) == 0) {

throw string("Что-то пошло не так");

}

string ans(32, '0');

SetSign(ans, S);

SetExp(ans, E);

SetMant(ans, M);

Print("a \* b", ans);

return;

}

void Multiplication(float a, float b) {

Multiplication(String(a), String(b));

}

Листинг 5 ­– Файл «Division.h»

#pragma once

#include <iostream>

#include <string>

#include "Helpers.h"

using namespace std;

static void Division(string a, string b) {

cout << "Деление\n";

Print("a", a);

Print("b", b);

if (IsZero(b)) {

throw string("Деление на ноль");

}

if (IsZero(a)) {

cout << "a / b = 0\n";

return;

}

bool SignBitA = SignBit(a), SignBitB = SignBit(b);

int SignA = Sign(a), SignB = Sign(b);

int ExpA = Exp(a), ExpB = Exp(b);

int MantA = Mant(a), MantB = Mant(b);

int S = SignA \* SignB;

int E = ExpA - ExpB;

long long M = ((long long)MantA << 23) / MantB;

if (E >= 128) {

throw string("Переполнение порядка");

}

if (E <= -127) {

throw string("Потеря значимости порядка");

}

if (M == 0) {

throw string("M = 0 при делении");

}

// Нормализация

while (M >= (1 << 24)) {

throw string("Что-то пошло не так");

}

if ((M & (1 << 23)) == 0) {

M <<= 1;

E--;

if (E <= -127) {

cout << endl;

throw string("Потеря значимости порядка");

}

}

string ans(32, '0');

SetSign(ans, S);

SetExp(ans, E);

SetMant(ans, M);

Print("a / b", ans);

return;

}

static void Division(float a, float b) {

Division(String(a), String(b));

}

3 РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

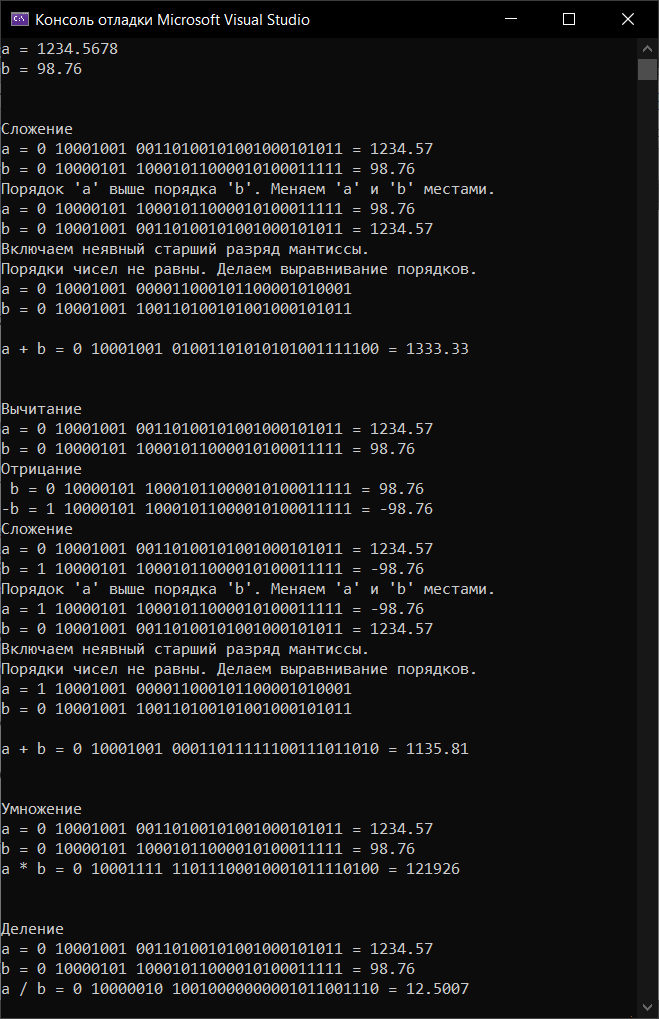


Рисунок 1 – Результат выполнения программы

4 ВЫВОД

Таким образом, была создана программа эмулятора АЛУ, выполняющего операции сложения, вычитания, умножения и деления над двумя числами с плавающей точкой. На ввод программа принимает два числа, представленных в десятичной форме.