|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 «ДЛИННАЯ АРИФМЕТИКА»**

Студент Светличная Алина Алексеевна

Группа ИУ7 – 33Б

Проверил Силантьева Александра Васильевна

*2021 г.*

# **Описание технического задания**

Входные данные:

Действительное число: строка, содержащая вещественное число в экспоненциальном виде <±m.nE±K> или в стандартном виде <±m.n>.

Целое число: строка, содержащая целое число в стандартном виде.

Правила ввода вещественного числа:

• Мантисса может представляться с точкой и без нее: 12.34 или 1234 или 1234. или .1234

• Мантисса может содержать не более 30 цифр

• Порядок при вводе может быть или не быть: 1234 или 1234E0 или 12.34E2

• Порядок может содержать не более 5 цифр

• Регистр знака, отделяющего порядок от мантиссы, не важен: E или e

• Возможно отсутствие знака порядка и/или мантиссы, по умолчанию '+'

Правила ввода целого числа:

• Возможно отсутствие знака числа, по умолчанию '+'

Число может содержать не более 30 цифр

Выходные данные:

Длинное число в виде <±0.m1E±K1>. Длина мантиссы <m1> - до 30 цифр; длина порядка <K1> — до 5 цифр.

Действие программы:

Перемножение вещественного и целого чисел

.

Обращение к программе:

Запускается через терминал командой ./main.exe в директории с программой.

Возможные аварийные ситуации:

1. Некорректный ввод данных

* Пустой ввод вещественного числа
* Некорректные символы в мантиссе
* Некорректные символы в порядке
* В мантиссе более, чем одна точка
* Слишком большая длина мантиссы
* Слишком большая длина порядка
* Пустой ввод целого числа
* Некорректные символы в целом числе

1. Переполнение порядка при умножении

# **Описание структуры данных**

\* SIZE\_STRING\_FLOAT = 40, SIZE\_STRING\_INT = 32

**char num[SIZE\_STRING\_FLOAT]** – строка для считывания вещественного числа

**char num[SIZE\_STRING\_INT]** – строка для считывания целого числа

Поля структуры, описанной ниже, для хранения вещественного числа:

mantis\_sign – знак мантиссы

mantissa – мантисса числа

eps\_sign – знак экспоненты

eps\_num – значение порядка

point\_place – место точки в мантиссе

**typedef struct**

**{**

**char mantis\_sign;**

**char mantissa[MANTISSA\_MAX\_LEN];**

**char eps\_sign;**

**char eps\_num[EPS\_MAX\_LEN];**

**int point\_place;**

**} float\_number\_t;**

Поля структуры, описанной ниже, для хранения целого числа:

sign – знак целого числа

digit\_part – целочисленная часть

\*MANTISSA\_MAX\_LEN = 30, EPS\_MAX\_LEN = 5

**typedef struct**

**{**

**char sign;**

**char digit\_part[MANTISSA\_MAX\_LEN];**

**} int\_number\_t;**

\*MANTISSA\_MAX\_LEN = 30

**int result[MANTISSA\_MAX\_LEN \* 2] = {0} –** массив для хранения результата умножения

# **Описание алгоритма**

\*Функции закомментированы в программе

В программе использован классический алгоритм умножения “столбиком”. Берется вторая строка, которая посимвольно проходится от своего конца до начала (т.е. из массива достается каждый символ, в котором хранится цифры числа), при каждой итерации умножая текущий символ (цифру) на первую строку, при этом складывая полученый результат умножения с “накопителем” (третья строка), в котором после полного прохода по числу и появится результат умножения.

# **ТЕСТЫ**

# **Аварийные тесты**

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

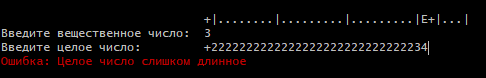
Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

* Изображение выглядит как текст

  Автоматически созданное описаниеИзображение выглядит как текст

  Автоматически созданное описаниеКоличество цифр превышает максимально возможное



* Изображение выглядит как текст

  Автоматически созданное описаниеПереполнение порядка
* Работа программы с корректными значениями

Изображение выглядит как текст

Автоматически созданное описание

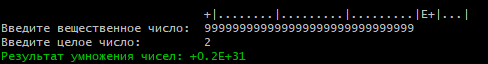
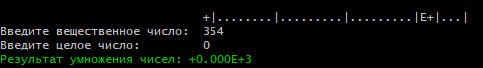
Изображение выглядит как текст

Автоматически созданное описание

* Изображение выглядит как текст

  Автоматически созданное описаниеИзображение выглядит как текст

  Автоматически созданное описаниеИзображение выглядит как текст

  Автоматически созданное описаниеРабота программы со знаками
* Округление
* Умножение на 0

# **Контрольные вопросы**

1. Каков возможный диапазон чисел, представляемых в ПК?

Максимальное значение беззнакового целого числа, для которого выделяется 64 разряда, равно 2 в 64 степени (18 446 744 073 709 551 615).

Диапазон чисел зависит от выбранного типа, разрядности процессора и памяти, выделенной для хранения числа.

2. Какова возможная точность представления чисел, чем она определяется?

Память, выделяемая под хранение мантиссы числа, определяет точность представления вещественных чисел. Для мантиссы числа типа double выделяется 52 бита, с помощью этого мантисса числа может иметь значение до 4 503 599 627 370 496.

3. Какие стандартные операции возможны над числами?

Операции сложения, вычитания, умножения, деление, взятие остатка (только для целых чисел), сравнение.

4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?

Для этого можно использовать массив, в котором каждый элемент будет храниться в своем месте или структуру данных, в которой можно каждую часть числа хранить в определенной переменной (в лабораторной работе был смоделирован второй вариант).

5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Для этого можно воспользоваться библиотечными функциями (например, библиотека gmp) или собственноручно написать необходимые процедуры.

Например, в данной работе была использована структура для хранения каждой цифры в отдельном массиве и некоторые другие параметры, описанные выше, написана функция, моделирующая умножение в столбик (аналогичным способом можно делить «длинные цифры»).

# **Вывод**

При написании лабораторной работы я познакомилась с длинной арифметикой. Что помогло мне понять, как располагаются числа в памяти компьютера, как происходит переполнение чисел и научиться обходить ограничение языка программирования, создавая свои собственные операции для рабты с такими числами.