|  |  |
| --- | --- |
| Изображение выглядит как герб, эмблема, символ, нашивка  Автоматически созданное описание | **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 ПО ДИСЦИПЛИНЕ:**

**ТИПЫ И СТРУКТУРЫ ДАННЫХ**

**ТЕМА: "ДЛИННАЯ АРИФМЕТИКА": ОБРАБОТКА БОЛЬШИХ ЧИСЕЛ**

Студент **Поздышев А. В.**

Группа **ИУ7-31Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

|  |  |
| --- | --- |
| Студент | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Поздышев А. В.** |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Силантьева А. В.** |

**2024**

Оглавление

[ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ 2](#_Toc177987917)

[ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ 2](#_Toc177987918)

[ОПИСАНИЕ ВНУТРЕННИХ СТРУКТУР ДАННЫХ 3](#_Toc177987919)

[ФУНКЦИИ ПРОГРАММЫ 3](#_Toc177987920)

[ОПИСАНИЕ АЛГОРИТМА 4](#_Toc177987921)

[НАБОР ТЕСТОВ 5](#_Toc177987922)

[ВЫВОДЫ 7](#_Toc177987923)

[ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ 7](#_Toc177987924)

## ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Смоделировать операцию деления действительного числа на действительное число в форме [±]m[.]n[Е[±]K], где суммарная длина мантиссы (m+n) - до 40 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме ±0.m1 Е ±K1, где m1 – до 40 значащих цифр, а K1 - до 5 цифр.

## ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Входные данные

Две строки, содержащие действительные числа в экспоненциальной или обычной форме.

Действительное число – строка вида <[±]m[.]n[E[±]K]>, где длина мантиссы (m+n) – до 40 значащих цифр, а K – до 5 цифр.

Выходные данные

Строка, содержащая в себе результат деления двух введенных действительных чисел.

Результат деления двух действительных чисел представляется в виде <±0.m1E±K1>, где m1 – мантисса до 40 значащих цифр, а K1- до 5 цифр.

Описание задачи, реализуемой программой

Задача программы – осуществление деления одного действительного числа на другое действительное число.

Способ обращения к программе

Ввод и вывод всех данных осуществляется через консоль.

Описание возможных аварийных ситуаций и ошибок пользователя

Аварийные ситуации:

* Пустое поле ввода (ожидание ввода пользователя);
* Переполнения порядка при делении;
* Деление на ноль;

Ошибки пользователя:

* Некорректный ввод: превышение допустимой длины мантиссы (m+n);
* Некорректный ввод: превышение допустимой длины значения K;
* Некорректный ввод: посторонние символы в записи числа;
* Некорректный ввод: запись числа, не подходящая маске <[±]m[.]n[E[±]K]>;

# ОПИСАНИЕ ВНУТРЕННИХ СТРУКТУР ДАННЫХ

Введенные действительные числа в программе представляются в структурном типе big\_num.

Листинг структуры big\_num:

|  |
| --- |
| #define LEN\_MANTISA 41  struct big\_num  {      int massive[LEN\_MANTISA];      int point\_pos;  size\_t len;      char sign;      int power;  }; |

1. massive – массив целых чисел, элементами которого являются цифры мантиссы введенного действительного числа.
2. point\_pos – индекс позиции точки в действительном числе.
3. len – длина мантиссы введенного числа.
4. sign – знак введенного действительного числа.
5. power – значение порядка введенного действительного числа.

# ФУНКЦИИ ПРОГРАММЫ

Complex\_devide

Заголовок:

|  |
| --- |
| int complex\_devide(struct big\_num \*num\_1, struct big\_num \*num\_2, struct big\_num \*result); |

Функция осуществляет деление действительных чисел типа big\_num. Во время ее выполнения она вызывает такие функции как: simple\_multy и to\_sub. Функция принимает указатели типа big\_num на делимое num\_1, делитель num\_2, частное result. Возвращает код ошибки выполнения операции.

Simple\_multy

Заголовок:

|  |
| --- |
| int simple\_multy(struct big\_num \*num, int multy); |

Функция осуществляет умножение действительного числа типа big\_num на однозначное типа int. Функция принимает указатель типа big\_num на число num, однозначное число multy. Возвращает код ошибки выполнения операции.

To\_sub

Заголовок:

|  |
| --- |
| int to\_sub(struct big\_num \*num\_1, struct big\_num \*num\_2); |

Функция осуществляет вычитание целых чисел типа big\_num. Функция принимает указатели типа big\_num на уменьшаемое num\_1, вычитаемое num\_2. Разность записывается в num\_1. Возвращает код ошибки выполнения операции.

Take\_num

Заголовок:

|  |
| --- |
| int take\_num(char const \*buffer, struct big\_num \*num); |

Функция считывает из буфера число структурного типа big\_num. Функция принимает буфер buffer, указатель типа big\_num на число num. Возвращает код ошибки, если строка не подходит маске [±]n[.]m[E[±]k], где (n + m) = 40 до 40 цифр, k до 5 цифр.

Print\_num

Заголовок:

|  |
| --- |
| void print\_num(struct big\_num num); |

Функция выводит число структурного типа big\_num на экран в виде ±0.mEk, где m до 40 цифр. Функция принимает число num типа big\_num.

Print\_err

Заголовок:

|  |
| --- |
| void print\_err(int const rc); |

Функция выводит сообщение об ошибке. Функция принимает код ошибки rc.

# ОПИСАНИЕ АЛГОРИТМА

Программа предлагает пользователю ввести два действительных числа. После введения строк программа считывает их и проверяет каждую из них на валидность. Если строки не корректны, то выводится сообщение об ошибке ввода.

Во время выполнения алгоритма в начале оба действительных числа приводятся к целому ввиду за счет изменения их порядков и смещения позиции точки. На втором шаге находится неполное делимое большее делителя за счет сравнивания старших разрядов неполного делимого и делителя. После нахождения неполного делимого младшие разряды делителя и неполного делимого откидываются до (k – 1), где k длина делителя. В результате получаем оба числа t\_1 и t\_2. На следующем шаге после нахождения частного t = t\_1 / t\_2 происходит сравнивание произведения числа t на делитель и неполного делимого. Если произведение оказывается больше неполного делимого, число t уменьшается на 1 и сравнивание повторяется, иначе на последующем шаге происходит вычитание неполного делимого и делителя, результат выполнения которых сохраняется в качестве нового неполного делимого. Число t сохраняется в качестве новой цифры частного. Далее повторяется алгоритм со 2 шага до тех пор, пока размер мантиссы частного не станет максимальным или результат вычитания неполного делимого и делителя не станет 0 на последней итерации разрядов делимого. На последней итерации разрядов делимого сохраняется позиция точки в частном.

Результат деления двух чисел выводится в виде <±0.m1E±K1>, где m1 до 40 цифр, а K1 до 5 цифр.

# НАБОР ТЕСТОВ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название | Число №1 | Число№2 | Результат |
| 1 | Некорректный ввод | abc | - | ERR\_IO: Ошибка ввода. |
| 2 | Некорректный ввод | +abc.defe+10 | - | ERR\_IO: Ошибка ввода. |
| 3 | Некорректный ввод | e+10 | - | ERR\_IO: Ошибка ввода. |
| 4 | Кол-во цифр в мантиссе больше 40 | 999999999999999999999.99999999999999999999e1 | - | ERR\_IO: Ошибка ввода. |
| 5 | Кол-во цифр в порядке больше 5 | 123.123e123456 | - | ERR\_IO: Ошибка ввода. |
| 6 | Порядок – не целое число | 123.123e3.14 | - | ERR\_IO: Ошибка ввода. |
| 7 | Некорректный ввод | 15.53.35 | - | ERR\_IO: Ошибка ввода. |
| 8 | Деление на ноль | 123.123e10 | 0 | ERR\_DATA: Ошибка данных |
| 9 | Деление нуля на ноль | 0 | 0 | ERR\_DATA: Ошибка данных. |
| 10 | Деление нуля на число | 0e1 | 123.123e10 | 0.0e0 |
| 11 | Деление числа на себя | 123.123e10 | 123.123e10 | +0.1e1 |
| 12 | Деление числа на число | 121 | 11 | +0.11e2 |
| 13 | Деление максимального числа на минимальное | +9999999999999999999999999999999999999999e+99999 | -9999999999999999999999999999999999999999e-99999 | ERR\_NULL: Переполнение данных. |
| 14 | Переполнение порядка | 123.123e5000 | 123.123e-5000 | ERR\_NULL: Переполнение данных. |
| 15 | Округление в большую сторону | 2 | 3 | +0.6666666666666666666666666666666666666667e0 |
| 16 | Округление в меньшую сторону | 1 | 3 | +0.3333333333333333333333333333333333333333e0 |
| 17 | Деление чисел с разными знаками | +15 | -2 | -0.75e1 |
| 18 | Перенос округления на старшие разряды | 9999999999999999999999999999999999999999 | 2 | +0.5e40 |

## ВЫВОДЫ

Для представления больших действительных чисел в памяти компьютера очень удобно использовать такой тип данных как структура. С помощью структуры можно разделить число на его компоненты, что значительно упрощает хранение и дальнейшую работу с ним.

# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1.Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от размера области памяти, выделенной компьютером. Например, тип int занимает 4 байт – диапазон от -2 147 483 648 до 2 147 483 647 для 64-разрядных ПК.

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

Вещественные числа хранятся в представлении с плавающей точкой в виде X = M \* n^p, где M – мантисса в виде дроби. Точность числа зависит от максимально возможной длины это мантиссы, которая зависит от области выделенной памяти (кол-во бит). При выходе за длину мантиссы происходит округление. Для n-битного целого числа максимальная представленность (максимальное значение) определяется формулой: 2^n – 1.

Для 64 разрядов это 2^64 – 1 = 18 446 744 073 709 551 615.

3.Какие стандартные операции возможны над числами?

Стандартные операции над числами – сложение, вычитание, умножение, деление.

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

Для представления больших чисел в памяти компьютера можно использовать структуры, в которой будет храниться мантисса, порядок, знак, цифры числа и другое.

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

Операции над большими числами выполняются путем последовательного выполнения арифметических операций над их цифрами, которые хранятся массиве.