|  |  |
| --- | --- |
| 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 Иванов Павел Александрович

Группа ИУ7 – 35Б

Вариант №6

**Оглавление**

[Описание условия задачи 2](#_Toc82421391)

[Описание технического задания 2](#_Toc82421392)

[Описание структуры данных 4](#_Toc82421393)

[Описание алгоритма 4](#_Toc82421394)

[Набор тестов 4](#_Toc82421395)

[Ответы на контрольные вопросы 8](#_Toc82421396)

[Вывод 9](#_Toc82421397)

# **Описание условия задачи**

Смоделировать операцию деления действительного числа в форме ±m.n Е ±K, где суммарная длина мантиссы (m+n) - до 30 значащих цифр, а величина порядка K - до 5 цифр, на целое число длиной до 30 десятичных цифр. Результат выдать в форме ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

# **Описание технического задания**

Входные данные:

Действительное число: строка, содержащая вещественное число в виде “+\-m.nE+\-K”. Знак перед числом и перед порядком обязательно. Знак экспоненты “E” необязательно. Суммарная длина (m+n) - до 31 знака, включая точку; длина порядка — до 5 цифр.

Целое число: строка, содержащая целое число в виде “+\- m”. Знак перед числом обязательно. Длина модуля числа “m” - до 30 цифр.

Выходные данные:

Длинное число (результат деления) в виде “+\-0.m1E+\-K1”. Длинна мантиссы “m1” - до 30 цифр; длинна порядка “K1” — до 5 цифр.

Действие программы:

Деление действительного числа на целое.

Обращение к программе:

Программа запускается через терминал.

Аварийные ситуации:

1. Ввод пустых данных

На выходе сообщение: «No data! Try again!»

1. Ввод чисел в неправильном формате (присутствуют лишние символы, или в числе две запятые, или в числе две «Е», или в целом числе есть не только цифры, или отсутствует знак)

На выходе сообщение: «Invalid input!»

1. Порядок действительного числа превышает максимально допустимый или мантисса превышается 30 чисел

На выходе сообщение: «Too big floating number!»

1. Вводимое целое число слишком длинное

На выходе сообщение: «Too big integer number!»

1. В поле для целого числа введён ноль

На выходе сообщение: «Division by zero is prohibited!»

1. При вычислениях достигнут машинный ноль

На выходе сообщение: «Machine zero reached!»

# **Описание структуры данных**

Для хранения длинного числа я использовал структуру

struct long\_num

{

char sign\_mant;

char mantissa[MANTISSA\_SIZE + 1];

int exp;

};

Она подходит для хранения как вещественных, так и целых чисел.

sign\_mant – знак мантиссы, занимает 1 байт (символ)

mantissa – массив символов, хранящий мантиссу

exp – порядок, целое число

# **Описание алгоритма**

1. Считывание числа и проверка на корректность.
2. Мантисса вещественного числа приводится к виду без запятой.
3. К мантиссе вещественного числа добавляются нули, пока она не станет больше целого числа.
4. Все добавки нулей и число цифр после запятой запоминается, чтобы потом вычесть
5. Выполняется простое деление уголком: каждый раз подбирается максимально возможное число и запоминается
6. Число нормализируется, степень числа 10 восстанавливается

# **Набор тестов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тестовый случай | Действительное число | Целое число | Результат |
| 1. | Обычный тест с положительными числами | +14.2 | +2 | +0.71E+1 |
| 2. | Числа разных знаков | +2.12E+1 | -4 | -0.53E+1 |
| 3. | Числа разных знаков | -2.16E+1 | +4 | -0.54E+1 |
| 4. | Обычный тест с отрицательными числами | -16.15E+0 | -5 | +0.323E+1 |
| 5. | Обычный тест на больших числах | +52.3E+45 | +123456789123456789 | +0.423630003431403031649397587586E+30 |
| 6. | Обычный тест на маленьких числах | +5.7E-45 | -123456789 | -0.461700004201470038233377347924E-52 |
| 7. | Крайнее значение целого числа | +5E+30 | +999999999999999999999999999999 | 0.500000000000000000000000000001E+1 |
| 8. | Крайнее значение действительного числа | +99999999999999999999E+99999 | +100000000000000000000000 | +0.99999999999999999999E+99996 |
| 9. | Первое число – абсолютный ноль | +0E+45 | +12 | +0.0E+0 |
| 10. | Первое число – абсолютный ноль | -0E-10 | +12 | +0.0E+0 |
| 11. | Первое число – абсолютный ноль | +0 | +12 | +0.0E+0 |
| 12. | Первое число – абсолютный ноль | -0 | +12 | +0.0E+0 |
| 13. | Первое число без ведущего нуля | +.456 | +2 | +0.228E+0 |
| 14. | Первое число без десятичной части | +456 | +2 | +0.228E+3 |
| 15. | Первое число без десятичной части | +456. | +2 | +0.228E+3 |
| 16. | Деление на ноль | +42 | +0 | Division by zero is prohibited! |
| 17. | Деление на ноль | -87 | -0 | Division by zero is prohibited! |
| 18. | Деление на ноль | +0E+0 | +0 | Division by zero is prohibited! |
| 19. | Пустой ввод |  |  | No data! Try again! |
| 20. | Неправильный ввод | 3E+4 | +2 | Invalid input! |
| 21. | Неправильный ввод | 1 | +2 | Invalid input! |
| 22. | Неправильный ввод | ++3E4 | +123 | Invalid input! |
| 23. | Неправильный ввод | +3E--4 | +123 | Invalid input! |
| 24. | Неправильный ввод | +abc | +123 | Invalid input! |
| 25. | Неправильный ввод | +3E+3E+3 | +123 | Invalid input! |
| 26. | Неправильный ввод | +3.452.4E+4 | +123 | Invalid input! |
| 27. | Неправильный ввод | +3.452E+4.2 | +123 | Invalid input! |
| 28. | Неправильный ввод | +3.4+56E+7 | +123 | Invalid input! |
| 29. | Неправильный ввод | + | +123 | Invalid input! |
| 30. | Неправильный ввод | - | +123 | Invalid input! |
| 31. | Неправильный ввод | +123 | + | Invalid input! |
| 32. | Неправильный ввод | +123 | - | Invalid input! |
| 33. | Неправильный ввод | +123 | +423.12 | Invalid input! |
| 34. | Неправильный ввод | +123 | 120 | Invalid input! |
| 35. | Неправильный ввод | +123 | ++432 | Invalid input! |
| 36. | Неправильный ввод | +123 | abc | Invalid input! |
| 37. | Неправильный ввод | +123 | +42E+10 | Invalid input! |
| 38. | Слишком большое действительное число | +3E+999999 | +123 | Too big floating number! |
| 39. | Слишком большое действительное число | +12345678901234567890123456E+99999 | +123 | Too big floating number! |
| 40. | Слишком большое целое число | +123 | +1234567890123456789012345678901 | Too big integer number! |
| 41. | Переполнение разрядной сетки | +99999999999999999999E+99999 | +10000000000 | Machine Zero Reached! |
| 43. | Округление 9-тки | +10.9999999999999999999999999999 | +1 | +0.73839837398373983739837398374E+2 |
| 44. | Округление последнего числа | +2 | +3 | +0.666666666666666666666666666667E+0 |
| 45. | Циклическое округление | +999999999999999999999999999999 | +2 | +0.5E+30 |

# **Ответы на контрольные вопросы**

*1. Каков возможный диапазон чисел, представляемых в ПК?*

Диапазон чисел зависит от разрядности системы. На 64-разрядных процессорах невозможна одновременная обработка более чем 64 бит (8 байт). Поэтому максимальное возможное число на таком ПК 264-1. В общем случае, максимальное возможное число определяется как 2n-1, где n — разрядность системы.

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

Для целых чисел вопрос точности не стоит. По-иному обстоит дело с числами вещественными. Точность вещественного числа определяется длиной мантиссы. Соответственно, чем больше разрядом выделено мантиссу, тем точнее число.

Например, для мантиссы числа типа double выделяется 52 бита, с помощью этого мантисса числа может иметь значение до 4 503 599 627 370 496. Если мантисса выходит за разрядную сетку, то происходит округление.

*3. Какие стандартные операции возможны над числами?*

Над числами возможны операции сложения, деления, умножения, вычитания, взятие целой части, взятие остатка.

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

Для обработки подобных случае удобно использовать массивы символов или числовые массив. Также можно использовать пользовательские структуры, например, храня в одной структуре число «по частям».

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

Требуется самостоятельно разработать функции для обработки таких операций, выбрав подходящий алгоритм в зависимости от выбранной структуры данных (для числа, разделенного на несколько, или числа, разделенного в массив символов). Например, для числа, разделенного на символы, можно написать классический алгоритм деления «в столбик».

# **Вывод**

В ходе данной работы я осознал, что для работы с большими числами программист должен самостоятельно писать программы для их обработки. Большие числа могут быть представлены разными способами. Например, массивом символов или с помощью нескольких меньших чисел. В зависимости от выбранной структуры можно подобрать алгоритм, который будет проделывать желаемые операции.

Я научился работать с длинной арифметикой, лучше разобрался, как хранятся числа в компьютере. Понял области применения длинной арифметики и разработал программу, которая умеет с ней работать. Также я выяснил, что обработка длинных чисел — непростая задача, порой требующая как временных, так и физических затрат.