



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

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Название предмета: Типы и структуры данных

Студент: Варламова Екатерина Алексеевна

Группа: ИУ7-31Б

2019г.

I. Описание условия задачи

Смоделировать операцию деления целого числа длиной до 30 десятичных цифр на действительное число в форме $\pm m.n \text{ E } \pm K$, где суммарная длина мантиссы ($m+n$) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $\pm 0.m1 \text{ E } \pm K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр. Программа должна осуществлять ввод чисел и выдавать либо верный результат в указанном формате (при корректных данных), либо сообщение о невозможности произвести счет.

II. Техническое задание

1. Описание исходных данных

Вводится 2 числа: целое и вещественное. При этом первое введенное число считается делимым, а второе – делителем.

Формат ввода:

Для действительных чисел допускается ввод в обычном формате (например, 0.025, +.0123, -.78), а также с указанием порядка (например, 1.9e-67, 23.E78).

Не допускается постановка пробелов между любыми двумя символами в числе и использование запятой вместо точки.

Ограничения:

- длина целого числа больше 0 и не превышает 30;
- длина мантиссы вещественного числа больше 0 и не превышает 30;
- порядок вещественных чисел находится в диапазоне: [-99999; 99999];

2. Описание результата программы

При корректных данных будет выдан результат деления двух введенных чисел. В ином случае будет выдано сообщение об ошибке (например, о делении на ноль, о неверном формате данных).

Формат вывода:

$\langle \text{знак} \rangle 0.\langle \text{мантисса} \rangle \text{E} \langle \text{знак порядка} \rangle \langle \text{порядок} \rangle$

При этом знак у порядка и у мантиссы выводится только тогда, когда имеет значение “-”.

Ограничения:

- длина мантиссы результата не превышает 30. Если результат деления не может быть записан в 30 значащих цифр, то он округляется;
- порядок результата находится в диапазоне: [-99999; 99999]. Если порядок результата превышает максимальное значение, выводится слово “infinity”. Если порядок результата меньше минимального значения, выводится “0”.

3. Описание задачи, реализуемой программой

Программа выполняет деление целого числа на вещественное с учётом ограничений, описанных выше. Результат деления нормализуется и выводится строго в указанном выше формате. При некорректных данных выводится сообщение об ошибке.

4. Способ обращения к программе

Необходимо установить компилятор (например, gcc-9.2) и осуществить запуск программы из командной строки. Дальнейшие инструкции будут выведены после запуска.

5. Описание возможных аварийных ситуаций и ошибок пользователя

- совершение операции не имеет смысла (деление на 0);
- входные данные выходят за рамки ограничений;
- входные данные являются бессмысленными с точки зрения задания;

Во всех указанных случаях программа завершится корректно и сообщит об ошибке.

III. Описание внутренних структур данных

Как известно, на 64-разрядных процессорах максимально под представление мантиссы действительного числа отводится 52 разряда, а под представление порядка – 11 разрядов. С учётом таких ограничений максимальное число, которое можно записать в стандартный числовой

(вещественный) тип данных, содержит примерно 20 десятичных разрядов. Поскольку по условию задачи требуется производить вычисления с числами большей точности (30 десятичных разрядов), использование стандартных числовых типов данных становится невозможным. В связи с этим, принято решение организовать структуру для хранения всего числа, в качестве полей которой выступают: знак мантиссы как переменная символьного типа, мантисса в виде массива коротких целых, длина мантиссы в виде короткого целого и порядок как целое знаковое число. При выборе структуры данных, лучше всего подходящей под хранение мантиссы, мы прежде всего руководствовались тем, чтобы максимально приблизиться по размеру занимаемой памяти к представлению чисел в стандартных типах данных. Именно поэтому массив является наиболее предпочтительным вариантом хранения. Для контроля ввода пользователя используется массив символьного типа. После обработки введённых данных корректная информация записывается в описанную выше структуру.

IV. Описание алгоритма

Вся информация, введённая пользователем, помещается в массив символьного типа. Этот массив обрабатывается таким образом, чтобы выделить мантиссу, определить порядок и знак. Если обработка прошла успешно, информация помещается в структуру, иначе выдается ошибка о неверном формате входных данных. Далее проверяется деление на ноль: если мантисса вещественного числа содержит только нули, то выдается сообщение о соответствующей ошибке. Затем осуществляется операция деления с помощью алгоритма «в столбик», то есть циклически выполняются следующие действия:

- вычитание делителя из части делимого до тех пор, пока возможно, и подсчёт количества успешных вычитаний (для этого так же используется имитация вычитания в столбик);

- запись количества вычитаний в мантиссу результата (не учитывая незначащие нули);

- если в делимом больше нет цифр, а остаток от деления есть, то к делимому дописывается ноль и уменьшается порядок результата;

Указанная последовательность действий повторяется до тех пор, пока в мантиссе результата есть место или до тех пор, пока размер массива с делимым позволяет дописывать нули. По завершении определяется знак результата и его порядок, в случае необходимости округляется мантисса (с учётом циклического переноса). Данный алгоритм является предпочтительным при выбранной структуре данных, ведь массив в первую очередь обеспечивает удобное и быстрое обращение к составляющим его элементам, что очень полезно при вычислениях «в столбик».

V. Тестирование

Позитивные тесты:

- 01 - делитель представлен целым числом;
- 02 - делитель имеет вид: $-0<num>E<degree>$;
- 03 - делимое имеет вид: $+<num>.$, а делитель $<num>E<degree>$;
- 04 - при делении произошло переполнение (степень результата превысила максимальное значение);
- 05 - мантисса делимого максимально возможная по длине;
- 06 - обычный тест;
- 07 - при делении достигнут машинный ноль (степень результата ниже минимального значения);
- 08 - деление, при котором требуется округление (только последняя цифра увеличивается);
- 09 - деление, при котором требуется округление (причем последняя цифра мантиссы 9, а следующая больше 5, поэтому происходит перенос).

Негативные тесты:

- 01 - длина делимого превысила максимальное значение;
- 02 - делимое задано некорректно;
- 03 - длина делителя превысила максимальное значение;
- 04 - деление на число, мантисса которого не содержит чисел, отличных от нуля;
- 05 - делитель задан некорректно;
- 06 - порядок делителя меньше минимально возможного;
- 07 - порядок делителя превышает максимально возможный
- 08 - пустой файл
- 09 - отсутствует второе число.

VI. Выводы по проделанной работе

В результате проделанной работы было получено более полное представление о хранении вещественных чисел в современных компьютерах. Кроме того, работа с «большими» числами показала всю важность выбора наиболее удобной структуры данных для их представления.

VII. Ответы на вопросы

1. Возможные значения чисел находятся в диапазоне от $3.6 \text{ E } -4951$ до $1.1 \text{ E } +4932$ (при этом под мантиссу отводится 52 разряда, а под представление порядка – 11 разрядов).
2. Точность представления чисел определяется размером мантиссы.
3. Стандартные операции над числами: сложение (вычитание), умножение, деление.
4. Если обрабатываемые числа превышают возможный диапазон представления чисел в ПК, программист может использовать массив для хранения мантиссы и целое знаковое число для порядка.
5. Осуществить операции над числами, выходящими за рамки машинного представления, можно с помощью представления этих чисел в виде

массива и написания собственных функций над массивами, которые имитируют стандартные операции над числами.