

Лабораторная работа № 1

Мамврийский Иван ИУ-36Б

Описание условия задачи

Необходимо реализовать арифметические операции над числами, выходящими за разрядную сетку персонального компьютера, выбрать необходимые типы данных для хранения и обработки указанных чисел. Требуется смоделировать операцию деления действительное число на действительное число, где порядок имеет до 5 разрядов (от -99999 до 99999), а мантисса - до 30 знаков.

Описание ТЗ

Описание исходных данных и результатов:

(типы, форматы, точность, способ передачи, ограничения) Программа получает на вход два значения. Они вводятся в формате $\pm m.n E \pm K$, где суммарная длина мантиссы $m+k \leq 30$, а величина порядка K - до 5 цифр (т.е. порядок принимает значения от -99999 до +99999)

Результат выводится в формате $\pm 0.m E \pm K$, где m - мантисса до 30 значащих цифр, а K - порядок до 5 цифр

Описание задачи, реализуемой программой

Программа производит операцию деления первого введенного (действительного) числа на второе (действительное) и выводит результат в нормализованной форме, либо сообщает о невозможности произвести счёт.

Способ обращения к программе

Обращение к программе происходит путём консольного ввода чисел пользователем в заданном формате.

Описание возможных аварийных ситуаций и ошибок пользователя

Аварийные ситуации:

1. Результат деления не попадает под ограничения выводимого формата (происходит в случае, если абсолютное значение порядка превышает 99999)
2. Второй введенный параметр равен нулю (ошибка деления на ноль)
3. Ввод одного из параметров в некорректном формате (для первого параметра - попытка ввода вещественного числа, а также лишние символы в числе)
4. Ввод некорректных данных (нераспознаваемые символы в потоке ввода)

Описание внутренних СД

Основной тип, используемый в программе - структура.

```
typedef struct  
{  
    int first_sign_number; //наличие +(1) и -(0) до числа  
    int power_position; //расположение точки  
    int mantissa[MAX_LEN_MANTISSA]; //мантисса числа  
    int exp_position; //расположение exp  
    int second_sign_number; //наличие +(1) и -(0) после экспоненты  
    int order[MAX_LEN_ORDER]; //порядок числа  
} number_t;
```

В программе используются следующие конкретные реализации данного типа:

- **first_sign_number** - наличие знака
- **power_position** - позиция точки
- **Mantissa** - длиной 30
- **exp_position** - позиция экспоненты
- **second_sign_number** - наличие знака после экспоненты
- **Order** - порядок числа длиной 5

Описание алгоритма

Основные алгоритмы в программе - ввод числа и деление.

Ввод числа осуществляется путём считывания числа. Происходит его анализа. Далее идет распределение по данным структуры.

Деление числа происходит с использованием алгоритма деления в столбик

Основной принцип работы алгоритма: На входе действительные числа А и В, необходимо получить $C = A / B$. Пока числа А равно нулю, производим следующие действия: Пока $A \geq B$, вычитаем В из А и прибавляем единицу к С. Когда А становится меньше В, умножаем С и А на 10 (в представлении в виде цифр - сдвигаем влево на 1), контролируя выход за пределы представления вещественного числа.

Когда А станет равно нулю, завершаем. Итоговый результат, сохранённый в R будет мантиссой результата. Порядок же результата высчитывается как разность порядка А и порядка В.

Набор тестов с указанием проверяемого параметра

Ввод	Вывод	Что проверяется
99999999999999999999 9999999999 1	Result: +0.99999999999999999999 99999999999999999999e30	Деление порогового числа
0 2	Result: + 0.0 E + 0	числитель равный нулю
-150 и -0.3	Result: + 0.5 E + 3	деление отрицательного числа
1 30	Result: +0.33333333333333333333 33333333333333333333e-1	обработка округления
99999999999999999999 999999999999 и 1111111111111111111111 1111111111	Result: +0.9 E +1	деление с использованием предельно большого целого числа
99999999999999999999 999999999999 и 2	Result: +0.5 E +30	циклическое округление
10 100 E 99999	Result: +0.1 E -99999	наименьший возможный порядок
100 0	Result: No division by zero	обработка деления на ноль
43432.5325f	Result: Посторонние символы.	Проверка ввода некорректных данных
4135235.5246462.5325	Result: Введено неверное количество '.'.	Неверное количество точек
52346246.782738578937 85798237573287583728 57932759326572367567 83267865723109	Result: Неверная длина числа (m + n > 30).	слишком большое значение введённого числа
34134.5e135e	Result: Введено неверное количество	Неверно количество точек
4235325e324.4234	Result: Неверное расположение 'e' по отношению к '.'.	Неверное расположение e
100e1-1	Result: Неверное расположение знаков '+' или '-'.	Неверное расположение знака относительно e

Выводы

Для деления чисел больших размеров целесообразно использовать длинную арифметику. Она может быть получена путем представления чисел в виде массива цифр(соответствует числу в десятичной системе счисления). Для операций можно использовать классические алгоритмы, которые мы используем при подсчёте простых чисел.

Ответы на вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Целые числа (беззнаковые):

Выделенные разряды	Диапазон
16	0...65 535
32	0...4 294 967 295
64	0...18 446 744 073 709 551 616

Целые числа (со знаком):

Выделенные разряды	Диапазон
16	-32768...32767
32	-2 147 483 648...2 147 483 647
64	-9 223 372 036 854 775 808...9 223 372 036 854 775 807

Вещественные числа:

Выделенные разряды	Диапазон
32 (single precision)	3.4E-38...3.4E+38
64 (double precision)	1.7E-308...1.7E+308
80 (extended precision)	3.4E-4932...3.4E+4932

Диапазон значений вещественного числа ограничен длиной порядка мантисы

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

Возможная точность определяется в случае вещественных чисел длиной мантиссы числа, для 32-битного вещественного числа с мантиссой длиной 23 бита, точность составляет 7-8 десятичных цифр, для 64-битного вещественного числа с мантиссой длиной 52 бита, точность составляет 15-16 десятичных цифр.

3. Какие стандартные операции возможны над числами?

Арифметические: сложение, вычитание, умножение и деление, унарный плюс и минус, инкремент и декремент.

Логические: сравнение.

Для целых: исключающее ИЛИ, логическое И, ИЛИ, побитовое отрицание

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

Для обработки таких чисел можно использовать массив цифр, используемый для представления вещественного чисел (с мантиссой, порядком) с основанием системы счисления 10.

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

Основные арифметические операции можно осуществлять, реализовав их подобно ручному счету, что будет довольно просто вследствие использования системы счисления с основанием 10.