**Министерство образования и науки 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 (ИУ)

КАФЕДРА: Программное обеспечение ЭВМ и информационные технологии (ИУ7)

**ОТЧЕТ ПО ЛАБРАТОРНОЙ РАБОТЕ**

**Лабораторная работа №1** по дисциплине

**“Типы и структуры данных”**

Обработка больших чисел

Студент: **Каримзай Ахмад-Халид ИУ7И-25Б**

Название предприятия: **МГТУ им. Н.Э. Баумана**

*2020 г.*

***Условие 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.

***Техническое задание***

*Исходные данные:*

На вход с консоли подаются две строки. Первая строка содержит в себе действительное число с мантиссой и порядком, вторая – целое число. Формат первой строки должен быть:

±\_\_\_\_до\_30\_цифр\_\_\_ E ±5цифр,

т. е. мантисса не должна содержать больше 30 цифр, а порядок – не больше 5; знак ± мантиссы и порядка при вводе не обязательны (тогда по умолчанию число положительно); написание порядка обязательно, разделять мантиссу и оператор следует буквой ‘E’ или ‘e’, обрамлённой пробелами, в мантиссе можно ввести десятичную точку (не запятую).

Формат второй строки должен соответствовать формату мантиссы первой, т. е. может не иметь знак (тогда по умолчанию число положительно) и содержит не больше 30 цифр.

Обе строки не могут содержать другие символы, кроме цифр, знаков + и – и, в случае вещественного числа, разделителя « E » или « e », десятичной точки.

*Результат*

Программа на выход на экран выдаёт нормализованную строку результата в формате:

±0. ±\_\_\_\_до\_30\_цифр\_\_\_ E ±5цифр,

т. е. формат совпадает с форматом первой входной строки.

*Описание задачи*

Программа выполняет деление длинных числе, которые не могут быть представлены в памяти компьютера с использованием стандартных типов данных, как единое число. По этой причине используются массивы для хранения цифр чисел.

*Аварийное завершение работы программы*

Программа завершается аварийно в случае, если:

* формат введённых строк не соответствует вышеуказанным;
* если на вход в качестве делителя был подан ноль, т. к. деление на ноль невозможно;
* превышение допустимого порядка числа при вычислениях и нормализации числа.

***Обращение к программе***

Исполняемый файл app.exe создается путем автоматической сборки проекта с помощью файла makefile. Для выполнения работы следует запустить данный исполняемый файл без каких-либо аргументов.

**Алгоритм:**

**1.** Ввод данных

**a)** Считывание

**b)** Проверка корректности

**c)** Нормализация

**2.** Деление

Проверяем степени двух чисел и, при необходимости, сдвигаем порядок.

До тех пор, пока кол-во полученных символов не равно 31 выполняем их нахождение:

Если делимое равно делителю, то текущий символ равен 1;

Если делимое меньше делителя, то производится сдвиг делителя на 1 позицию в лево (умножение на 10)

Если делимое больше делителя:

Пока делимое больше вычитаем делитель

счетчик++;

**3.** Вывод результата

а) Узнаем количество значащих чисел

б) Выводим значащие числа

в) Выводим порядок

**Структура данных:**

Для обработки данных используется структур num\_t где в этом структуре длинные числа хранятся в массивах типа short int. Где один элемент массива – цифра. Знак хранится в переменной типа int (-1 – отрицательное, 1 – положительное). Количество цифры хранятся в переменной типа int Порядок хранится в отдельной переменной типа int.

typedef struct

{

    int sign;

    int size;

    short int \*mant;

} num\_t;

**Функции:**

Функции ввода:  
В них заполняется массив, обрабатывается некорректность ввода

int input\_float(num\_t \*float\_t, int \*exponent);

int input\_int(num\_t \*int\_t);

Функции нормализации:

Убирают лишние нули, сдвигают массив для дальнейшей обработки.

void normalize\_int(num\_t \*int\_t);

void normalize\_float(num\_t \*float\_t, int dot, int \*exponent);

Функция count\_zero:

Считает количество нулей в начале массива (необходимо при нормализации)

int count\_zero(const num\_t \*input);

Функция move\_digits:

Сдвигает все элементы массива на заданное (num\_order) расстояние

void move\_digits(short int \*arr\_, const int size, int num\_order);

Функция check\_ten:

Проверяет не переходит ли результат деления за 10.

void check\_ten(short int \*arr\_float, short int \*arr\_int, int size, int \*exp);

Функции деления:

Сравнивает два массива. Возвращает res; -1 Значит, что a < b

int comparison(short int \*a, short int \*b);

Вычитает из массива a массив b

void subtract(short int \*a, short int \*b);

Производит операцию деления двух массивов

void division(short int \*arr\_float, short int \*arr\_int, short int \*result, int \*exp);

Функция округления массива по полследнему символу:

void rounding(short int \*arr, const int size, const int last\_digit);

Функция вывода результата:

void output\_res(const num\_t \*t, int exp);

***Тестовые данные***

|  |  |  |
| --- | --- | --- |
| Ввод | Вывод | Класс эквивалентности |
|  | Empty input! | Пустая строка |
| + | Empty input! | Нет мантиссы |
| E +11 | Empty input! | Нет мантиссы |
| +55 | Incorect Input: exponent overflow ( -100000, 100000 ) | Нет порядка |
| +55E11 | Nothing inputed! | Неверный формат порядка |
| +1234567890123456789012345678901E11 | Too many digits! | Введено больше 30 разрядов в мантиссу |
| +1 E +100001 | Incorect Input: exponent overflow ( -100000, 100000 ) | Переполнение порядка |
| -55 E -100001 | Incorect Input: exponent overflow ( -100000, 100000 ) | Нет порядка |
| +-1E+15 | Wrong Input! | Лишние знаки в мантиссе |
| +1 E +15 | Wrong Input! | Лишние пробелы |
| +1 E+12.5 | Wrong Input! | Вещественный порядок |
| +1 E+a | Wrong Input! | Символьный порядок |
| +1343a12E+15 | Wrong Input! | Недопустимый символ в мантиссе |
| 0.1E0  (пустая строка) | Nothing inputed! | Пустая строка |
| 0.1E0  + | Nothing inputed! | Нет цифр числа |
| 0.1E0  +1234567890123456789012345678901 | Wrong Input! | Введено больше 30 разрядов |
| 0.1E0  1a2312 | Wrong Input! | Недопустимые символы в строке |
| 0.1E0  1.2312 | Wrong Input! | Введена десятичная точка |
| 0.1E0  0 | Exception: Division by Zero. | Деление числа на 0 |
| 0E0  0 | Exception: Division by Zero. | Деление 0 на 0 |
| 0.0E55  1 | 0.0e0 | Деление 0 на число, отличное от 0 |
| +0.1E-99999  10 | Exponent overflow! | Переполнение после деления |
| +123.45E+01  1 | +0.12345e4 | Деление на 1 |
| +123.45 E +01  -1 | -0.12345e4 | Деление на -1 |
| +0.25e+01  100 | +0.25e-1 | Порядок меняет знак  c + на - |
| +0.25 E +01  10 | +0.25 E -00000 | Порядок становится нулевым |
| +2 E +00  3 | +0.666666666666666666666666666667E0 | После деления необходимо округление (доп. цифра > 4) |
| 1 E+00  3 | +0.333333333333333333333333333333e0 | После деления округление не нужно (доп. цифра <= 4) |
| +0.12 E-001  12 | +0.1e2 | Общий случай |
| +999999999999999999999999999999 E 0  2 | +0.5e30 | После деления необходимо округление (доп. цифра > 4), остаётся одна цифра |

**Контрольные вопросы:**

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

Диапазон значений чисел зависит от разрядности процессора, размера выделенной для хранения переменной памяти, от типа (целое/вещественное), знака. Под хранение целого положительного числа, для которого выделено 64 разряда, максимально возможное значение числа равно 2^64 – 1 = 18 446 744 073 709 551 615.

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

Точность представления вещественного числа зависит от максимально возможной длины мантиссы. Если длина мантиссы выходит за ганицы разрядной сетки, то происходит округление.

**3.** Какие стандартные операции возможны над числами?

Сложение, вычитание, умножение, деление.

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

Можно использовать специальные библиотеки для работы с большими числами; создать массив цифр числа и производить операции поэлементно, храня промежуточный результат в отдельном массиве; создать структуру, деля в ней мантису на части.

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

В данном случае возможно использовать поразрядные операции.

**Вывод:**

При реализации аримфетики длинных чисел целесообразно хранить цифры числа, как элементы массива. Арифметику производить поразрядно.