



(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 ПО ДИСЦИПЛИНЕ ТИПЫ И СТРУКТУРЫ ДАННЫХ

Группа **ИУ7-33Б**

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

Оценка

2022 г.

1. Описание условия задачи.

Смоделировать операцию деления действительного числа в форме $\pm m.n \text{ E } \pm K$, где суммарная длина мантииссы ($m+n$) - до 35 значащих цифр, а величина порядка K - до 5 цифр, на целое число длиной до 35 десятичных цифр. Результат выдать в форме $\pm 0.m1 \text{ E } \pm K1$, где $m1$ – округленная до 35 значащих цифр мантиисса, а $K1$ – порядок до 5 цифр.

2. Техническое задание.

Исходные данные и правила их ввода:

- Длинное вещественное число:
 - Длина строки с числом не превышает 150 символов.
 - Мантисса:
 - Количество значащих десятичных цифр мантиссы не превышает 35.
 - Перед мантиссой (или знаком и мантиссой) может быть указан знак + или - .
 - Перед мантиссой могут быть ведущие нули.
 - В записи числа может встретиться одна точка, разделяющая целую и дробную часть. Между точкой и ближайшими цифрами не должно быть пробелов.
 - Порядок:
 - Может быть указан порядок, длиной не более 5 цифр.
 - Перед значением порядка может быть указан знак + или - .
 - Значение порядка отделяется от числа латинской буквой e или E. Между порядком и мантиссой не может быть пробелов.
- Длинное целое число:
 - Длина строки с числом не превышает 150 символов.
 - Число:
 - Количество значащих десятичных цифр не превышает 35.
 - Перед числом может быть указан знак + или - .
 - Перед числом (или знаком и числом) могут быть ведущие нули.
- Результат:
 - В случае ошибки на экран выводится сообщение об ошибке в формате «ОШИБКА: описание ошибки».
 - Число:

- Вещественное число в формате $\pm 0.m \text{ E } \pm K$, где m – округленная до 35 значащих цифр мантисса результата деления, K – порядок результата деления до 5 значащих цифр.

Задача программы:

Выполнить деление длинного вещественного числа на длинное целое.

Способ обращения к программе:

Запуск исполняемого файла app.exe в консоли. Выполнить команду ./app.exe.

Возможные аварийные ситуации:

- Ошибка ввода строки:
 - Строка слишком большая.
 - Строка пустая.
- Ошибка преобразования строки в длинное вещественное число:
 - Строка содержит не валидные символы
 - Количество значащих цифр мантиссы превышает 35.
 - Длина порядка превышает 5 цифр.
- Ошибка преобразования строки в длинное целое число:
 - Встречен нечисловой символ или символы “+”, “-“ в неверном месте.
 - Количество значащих цифр превышает 35.
- Ошибки при делении:
 - Деление на ноль.

3. Описание внутренних структур данных.

Длинное вещественное число:

```
#define MANTISSA_SIZE 35
#define MANTISSA_CAPACITY MANTISSA_SIZE + 1
typedef struct
{
    int sign;
    char mantissa[MANTISSA_CAPACITY];
    size_t size;
    int exponent;
} big_float_t;
```

В структуре длинного вещественного числа содержатся 4 поля:

- `sign` - знак мантиссы - знаковое целое типа `int`. Во время выполнения программы принимает значения либо 0, либо 1, где 0 - число положительное, 1 - число отрицательное;
- `mantissa` - мантисса - массив беззнаковых целых размера 36. Значения элементов массива не принимают значения, более 10, поэтому был выбран тип `char`. Для выдачи результата с точностью в 35 знаков, нужно использовать одну цифру для возможного округления. Соответственно размер должен быть 36 цифр.;
- `size` - размер массива-мантиссы. Тип - `size_t`;
- `exponent` - порядок - знаковое целое типа `int`. Должно принимать значения от -99999 до 99999.

Длинное целое число:

```
#define DIGITS_SIZE 35
#define DIGITS_CAPACITY DIGITS_SIZE + 1
typedef struct
{
    int sign;
    char digits[DIGITS_CAPACITY];
    size_t size;
} big_int_t;
```

В структуре длинного целого числа содержатся 3 поля:

- `sign` - знак числа - знаковое целое типа `int`. Во время выполнения программы принимает значения либо 0, либо 1, где 0 - число положительное, 1 - число отрицательное;
- `digits` - само число - массив беззнаковых целых размера $35 + 1$. Значения элементов массива не принимают значения, более 10, поэтому был выбран тип `char`. При операции деления над длинным вещественным, из цифр его мантиссы будет создаваться длинное целое число, а по окончании операций, для возможного округления, нужно хранить 36 цифр;
- `size` - размер массива цифр, тип - `size_t`

Для работы со структурами используются следующие функции:

`int string_to_big_int(big_int_t *number, char *string);` - функция для преобразования строки в длинное целое число.

`int string_to_big_int(big_int_t *number, char *string);` - функция для преобразования длинного целого число в строку.

`int big_int_cmp(big_int_t *number1, big_int_t *number2);` - функция для сравнения двух длинных целых чисел.

`void big_int_copy(big_int_t *dst, big_int_t *src, size_t begin, size_t end);` - функция для копирования из одного длинного целого числа в другое.

`int big_int_is_zero(big_int_t *number);` - функция для проверки, является ли длинное целое число нулём.

`int big_int_subtract(big_int_t *minuend, big_int_t *subtrahend);` - функция для вычитания из длинного целого числа `minuend` длинное целое число `subtrahend`.

`int big_int_mult_int(big_int_t *big_int, int factor2);` - функция для умножения длинного целого на целое число типа `int`.

`int string_to_big_float(big_float_t *number, char *string);` - функция для преобразования строки в длинное вещественное число.

`int big_float_to_string(char *string, big_float_t *number);` - функция для преобразования длинного вещественного числа в строку.

`void big_float_round(big_float_t *number, size_t digits);` - функция для математического округления длинного вещественного числа до `digits` знаков.

`int big_float_div_big_int(big_float_t *quotient, big_float_t *dividend, big_int_t *divisor);` - функция для деления длинного вещественного числа `dividend` на длинное целое число `divisor` с сохранением результата в длинное вещественное `quotient`.

4. Описание алгоритма.

На вход программе подаются две строки. В первой должно содержаться длинное вещественное число. Из строки удаляются ведущие и конечные пробельные символы, происходит проверка на длину строки. В функции преобразования строки в длинное вещественное число число инициализируется (массив-мантисса заполняется нулями, поля знака, размера и порядка также принимают значения 0), далее происходит проверка корректности введённой строки и запись нужной информации в длинное вещественное число. Проверяется присутствие и правильность ввода знака: если указан знак “-”, поле `sign` принимает значение 1; если указан знак “+” или не указан вообще, поле `sign` не меняется. Игнорируются ведущие нули, значащие цифры последовательно записываются в массив-мантиссу, во время записи цифр происходит изменение размера массива-мантиссы и изменение порядка. Далее считывается значение порядка, если оно указано. Если при анализе строки количество значащих цифр превысило 35 или при анализе строки встречается некорректный символ, функция завершает работу и возвращает код ошибки, по которому далее выводится сообщение об ошибке. При полностью корректной строке функция корректно завершает свою работу и возвращает 0.

Во второй строке должно содержаться длинное целое число. Из строки удаляются ведущие и конечные пробельные символы, происходит проверка на длину строки. В функции преобразования строки в длинное целое число число инициализируется (массив цифр заполняется нулями, поле знака и размера также принимают значения 0), далее происходит проверка корректности введённой строки. Проверяется присутствие и правильность ввода знака: если указан знак “-”, поле `sign` принимает значение 1; если указан знак “+” или не указан вообще, поле `sign` не меняется. Игнорируются ведущие нули, значащие цифры последовательно записываются в массив, инкрементируется размер массива. Если размер массива превысил 35 или встретился некорректный символ, функция завершает работу и возвращает

код ошибки, по которому далее выводится сообщение об ошибке. При полностью корректной строке функция корректно завершает свою работу и возвращает 0.

После ввода и преобразования строк создается переменная для хранения длинного вещественного - результата деления, после чего запускается функция деления длинного вещественного на длинное целое.

Происходит подготовка к основному алгоритму. Делитель проверяется на 0 - если делитель - 0, функция завершается с ошибкой. Далее инициализируется длинное вещественное, в которое будет производиться запись результата. Мантисса делимого копируется в длинное целое. Далее делимое проверяется на 0. Если делимое - 0, в результат записывается 0 и функция завершается без ошибки. Далее определяется знак итогового значения при помощи операции xor над знаками делимого и делителя. В переменную результата записывается начальное значение порядка - разность между порядком делимого числа и порядком (длинной) делителя. Далее, пока мантисса делимого меньше делителя, делимое умножается на 10. Создаются переменные неполного делимого и переменная, в которой будет храниться временный делитель. Далее, по старшему разряду делимого и делителя определяется размер неполного делимого. Если старший разряд делимого больше старшего разряда делителя, длина неполного делимого равна длине делителя, если меньше, то длина неполного делителя на 1 больше, если равны, сравниваются следующие разряды. Далее происходит копирование найденного количества цифр из длинной целой копии мантиссы в переменную неполного делимого. Определяется индекс элемента целочисленной копии мантиссы, из которого впоследствии нужно будет “снести” новый разряд.

Начинается основной цикл. Цикл действует пока неполное делимое не равно нулю и пока количество найденных цифр мантиссы результата меньше размера массива-мантиссы.

Сначала происходит деление старшего разряда неполного делимого на старший разряд делителя. Если длина неполного делимого больше длины

делителя делятся два старших разряда. Получаем значение t . Далее делитель умножается на t и сравнивается с неполным делимым. Пока произведение делителя и t больше неполного делимого, произведение уменьшается на делитель, а t уменьшается на 1. Полученное t помещается в крайний элемент массива-мантиссы, размер мантиссы увеличивается. Далее полученное произведение вычитается из неполного делимого.

Если не достигнут конец делимого и неполное делимое меньше делителя, “сносится” следующий разряд и порядок результата увеличивается на 1. Если после “сноса” разряда неполное делимое остается меньше делителя, “сносится” следующий разряд, в крайний элемент массива-мантиссы записывается 0, значение порядка увеличивается на 1

Если достигнут конец делимого, а неполное делимое меньше делителя и не 0, к неполному делимому “сносится” 0. Если после “сноса” 0 неполное делимое остается меньше делителя, “сносится” ещё один 0, в крайний элемент массива-мантиссы записывается 0.

После вычисления нового значения неполного делимого, цикл повторяется.

По окончании цикла вызывается функция, которая уменьшает размер мантиссы до первого с конца ненулевого элемента. Функция завершается.

На экран выводятся введенные значения и значение результата в формате $\pm 0.m1 E \pm K1$. Результат округляется до 35 знаков. Если значения порядка вышло за пределы от -99999 до 99999, на экран выводится ошибка о переполнении.

5. Набор тестов.

№	Что проверяется	Вещественное число	Целое число	Вывод
Негативные тесты				
1	Пустая строка	-	-	ОШИБКА: пустая строка.
2	Нечисловой символ в записи вещественного числа	123a	-	ОШИБКА: При вводе числа встречается нечисловой символ.
3	Больше 35 значимых цифр мантиссы	1234567890123456789012.345678901234567890e-40	-	ОШИБКА: Превышено количество символов мантиссы.
4	Мантисса отделена от порядка не символом e или E	12.4 4	-	ОШИБКА: Ошибка в записи символа порядка.
5	При вводе порядка встречается нечисловой символ	2345.46E4g3	-	ОШИБКА: При вводе порядка встречается нечисловой символ.
6	Превышено количество символов порядка.	12E100000	-	ОШИБКА: Превышено количество символов порядка.
7	Деление на 0	1111111	0	ОШИБКА: Деление на

				ноль.
8	Превышение значения порядка.	1e99999	1	ОШИБКА: Превышено значение порядка.
9	Нечисловой символ при вводе целого числа	1	12e3	ОШИБКА: При вводе числа встречается нечисловой символ.
10	Количество значимых цифр целого превышает 35	1	83742563487 25637834827 35628374562 88822	ОШИБКА: Превышено количество символов.
Позитивные тесты				
11	Ввод вещественного числа без дробной части и без порядка.	12	3	+0.4 E1
12	Ввод вещественного числа с дробной частью без порядка	1.98	11	+0.18 E0
13	Ввод вещественного числа без дробной части с порядком с символом E.	14E2	1400	+0.1 E1
14	Ввод вещественного числа без дробной части с порядком с символом e.	132e6	11000	+0.12 E5
15	Ввод	21.873e5	30	+0.7291 E5

	вещественного числа с дробной частью и порядком.			
16	Ввод вещественного числа со знаком “-”	-12	3	-0.4 E1
17	Ввод вещественного числа со знаком “+”	+14	7	+0.2 E1
18	Ввод вещественного числа с ведущими нулями.	0000012	3	+0.4 E1
19	Ввод вещественного числа с ведущими нулями без числе до точки.	00000.00002	1	+0.2 E-4
20	Ввод вещественного числа со знаком “-” в порядке	5e-3	20	+0.25 E-3
21	Ввод вещественного числа со знаком “+” в порядке	5e+3	2000	+0.25 E1
22	Ввод 0	0	1	+0.0 E0
23	Ввод 0 со знаком “-”	-0	1	+0.0 E0
24	Ввод 0 со знаком “+”	+0	1	+0.0 E0
25	Ввод целого числа со знаком “-”	1	-50	-0.2 E-1

26	Ввод целого числа со знаком “+”	1	+50	+0.2 E-1
27	Ввод целого числа с ведущими нулями	1	00000001	+0.1 E1
28	Оба числа отрицательные	-4	-2	+0.2 E1
29	Результат - несократимая дробь	1	3	+0.3333333333 3333333333 3333333333 3333 E0
30	Результат - несократимая дробь с округлением до 35 знаков	2	3	+0.6666666666 6666666666 6666666666 6667 E0
31	Циклическое округление	+0.22223999999999 9999999999999999 999999 E35	2	+0.11112 E35

6. Вывод

Так как количество чисел, входящих в разрядную сетку среднестатистического ПК, ограничено, для работы с длинными числами нужно использовать готовые решения, либо реализовывать собственные, основанные на базовых математических операциях, доступных для чисел меньше, храня эти числа, например, поразрядно в массиве.

7. Контрольные вопросы.

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел может варьироваться в разных ПК, однако обычно максимальная занимаемая память для числа - 64 байта. Соответственно для беззнакового целого числа максимальным значением будет $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$.

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

Для хранения вещественного числа, память разбивается на три области: знак числа, порядок числа и мантисса. Обычно, в типе `double`, под мантиссу выделяется 52 бита, что означает 15-16 значащих десятичных разрядов

3. Какие стандартные операции возможны над числами?

Операции сложения, вычитания, умножения, деление, взятие остатка, сравнение.

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

В таком случае программист может воспользоваться готовыми решениями в области длинной арифметики, или же реализовать собственное. При реализации собственного программист может использовать стандартные целочисленные типы, так при помощи них можно представить длинное число как массив цифр, а также значение порядка и знака длинного числа.

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

Необходимо воспользоваться готовыми решениями, либо реализовать собственные функции для обработки длинных чисел, например, храня их поразрядно в целочисленном массиве.

