



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 **«ОБРАБОТКА БОЛЬШИХ ЧИСЕЛ»**

Студент _____ Хамзина Регина Ренатовна
фамилия, имя, отчество

Студент, группа

Хамзина Р.Р., ИУ7-33Б

2020 г.

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.

Если при делении чисел длина мантиссы стала больше 30 знаков, то необходимо произвести округление по следующему правилу: если 31-ая цифра больше или равна 5-ти, то число округляется по законам математики, иначе все последующие знаки отбрасываются.

Техническое задание

Входные данные

Целое число: строка, которая может состоять только из символов ‘+’/’-’ и цифр, в формате ‘+/-m’, где m – мантисса числа. Ввод знака перед числом обязателен. Длина мантиссы числа m меньше, либо равна 30 и больше 0.

Действительное число: строка, которая может состоять только из символов ‘+’/’-’, ‘.’, ‘E’ и цифр, в формате ‘+/-m[.n][E+/-k]’, где m – целая часть числа, n – дробная часть числа, ($m+n$) – мантисса числа, k – порядок числа. Ввод знака перед числом обязателен. Длина мантиссы числа ($m+n$) меньше, либо равна 30 и больше 0. Длина порядка k меньше, либо равна 5 и больше 0. Если число вводится с дробной частью n , то ввод точки после целой части m , ввод символа ‘E’, ввод знака порядка обязательны.

Выходные данные

Действительное число: строка, которая может состоять только из символов ‘+’/’-’, ‘.’, ‘E’ и цифр, в нормализованном формате ‘+-0.mE+/-k’, где m – мантисса числа, k – порядок числа. Длина мантиссы числа m меньше, либо равна 30. Длина порядка числа k меньше, либо равна 5.

Функция программы

Программа делит целое число на действительное.

Обращение к программе

Программа запускается из терминала командой «./app.exe» в директории с программой.

Возможные аварийные ситуации и ошибки пользователя

1. Некорректный ввод: строка с целым числом не содержит знака числа или первый символ не является знаком.

На выходе сообщение: «Первым символом целого числа должен быть знак числа.»

2. Некорректный ввод: строка с целым числом содержит не цифру в мантиссе.

На выходе сообщение: «Мантисса целого числа должна состоять только из цифр.»

3. Некорректный ввод: строка с целым числом содержит пробелы.

На выходе сообщение: «Не должно быть пробелов между символами.»

4. Неверная длина мантиссы целого числа.

На выходе: корректное число, длина мантиссы которого превышает 30 цифр.

На выходе сообщение: «Длина мантиссы целого числа должна быть меньше, либо равна 30 и больше 0.»

5. Некорректный ввод: строка с действительным числом не содержит знака числа или первый символ не является знаком.

На выходе сообщение: «Первым символом действительного числа должен быть знак числа.»

6. Некорректный ввод: строка с действительным числом содержит не цифру в мантиссе.

На выходе сообщение: «Мантисса действительного числа должна состоять только из цифр.»

7. Некорректный ввод: строка с действительным числом содержит пробелы.

На выходе сообщение: «Не должно быть пробелов между символами.»

8. Неверная длина мантиссы действительного числа.

На выходе: корректное число, длина мантиссы которого превышает 30 цифр.

На выходе сообщение: «Длина мантиссы действительного числа должна быть меньше, либо равна 30 и больше 0.»

9. Некорректный ввод: строка с действительным числом не содержит знака порядка или он не является знаком.

На выходе сообщение: «Перед порядком действительного числа должен быть знак порядка.»

10. Некорректный ввод: строка с действительным числом содержит не цифру в порядке.

На выходе сообщение: «Порядок действительного числа должен состоять только из цифр.»

11. Неверная длина порядка действительного числа.

На выходе: корректное число, длина порядка которого превышает 5 цифр.

На выходе сообщение: «Длина порядка действительного числа должна быть меньше, либо равна 5 и больше 0.»

12. Переполнение порядка.

На входе: в процессе деления степень полученного в результате числа превышает 99999.

На выходе сообщение: «Переполение порядка.»

13. Действительное число равно нулю.

На выходе сообщение: «Нельзя делить на 0.»

Внутренняя структура данных

Ввод: для ввода целого числа используется символьный массив *str[40]* который хранит все введенные символы числа. Для ввода вещественного числа используется символ *symbol* и массив *order[7]*.

Массив *str* разбивается на части, которые сохраняются в структуре *number*. При помощи *symbol* и *order* поля структуры *number* сразу заполняются.

```
typedef struct
{
    char number_sign;
    char mantissa[MAX_LEN_MANTISSA + 2];
    int point_index;
    char order_sign;
    int order;
} number;
```

Поля структуры:

- *number_sign* – знак числа. Принимает значения “+” или “-”;
- *mantissa*[*MAX_LEN_MANTISSA* + 2] – значение мантиссы числа, *MAX_LEN_MANTISSA* = 30, два дополнительных символа для знака и «\0»;
- *point_index* — индекс точки в числе;
- *order_sign* – знак числа. Принимает значения “+” или “-”;
- *int_order* – значение порядка числа.

В результате создаются две переменные структурного типа *number* *int_number* и *real_number*.

Деление: алгоритм деления позволяет записывать промежуточный результат в переменную структурного типа *number* *int_number*. Конечный результат записывается в переменную структурного типа *number* *result_number*.

В результате создается переменная структурного типа *number* *result_number*.

Вывод: для вывода используется созданная на этапе деления переменная структурного типа *number* *result_number*.

Алгоритм

1. Программа считывает две строки символов, одна из которых содержит целое число, а другая — вещественное число, и записывает полученные числа в переменные структурного типа *number*, одновременно проверяя на валидность.

```
int input_int_number(number *int_number);
```

```
int input_real_number(number *real_number);
```

2. Корректные числа преобразуются в нормализованный вид: удаляются ненужные нули, изменяется порядок числа в связи со сдвигом точки в начало числа.

```
void normalization(number *one_number);
```

```
void add_zeros(number *one_number);
```

3. Выполняется деление первого числа на второе методом “деления в столбик”, при этом контролируется округление, переполнение порядка числа и деление на ноль.

```
int compare_numbers(number *int_number, number *real_number);
```

```
int compare_zero(number *one_number);
```

```
void change_zero(number *one_number);
```

```
int subtraction(number *int_number, number *real_number);
```

```
int divide_numbers(number *int_number, number *real_number, number *result_number);
```

4. Корректный результат выводится на экран в нормализованном виде в соответствии со спецификацией, указанной в ТЗ.

```
void output_number(number real_number);
```

Функции программы

```
void print_conditions(void);
```

Описание: функция выводит назначение программы и условия ввода чисел.

Входные значения: -

Выходные значения: сообщения с информацией о назначении программы и правилах ввода чисел.

```
int input_int_number(number *int_number);
```

Описание: функция считывает целое число и записывает его в структуру *number int_number*.

Входные значения: переменная структурного типа *number int_number* для записи в нее числа.

Выходные значения: переменная структурного типа *number int_number* с записанным в нее числом, код ошибки.

```
int input_real_number(number *real_number);
```

Описание: функция считывает действительное число и записывает его в структуру *number real_number*.

Входные значения: переменная структурного типа *number real_number* для записи в нее числа.

Выходные значения: переменная структурного типа *number real_number* с записанным в нее числом, код ошибки.

void normalization(number *one_number);

Описание: функция нормализует мантиссу числа, записанную в *number one_number*, изменяя порядок числа.

Входные значения: переменная структурного типа *number one_number* с числом для нормализации.

Выходные значения: переменная структурного типа *number one_number* с нормализованным числом.

void add_zeros(number *one_number);

Описание: функция добавляет нули в структуру *number*.

Входные значения: переменная структурного типа *number*.

Выходные значения: переменная структурного типа *number* с добавленными нулями.

int compare_numbers(number *int_number, number *real_number);

Описание: функция сравнивает целое число *number int_number* и действительное число *number real_number*.

Входные значения: переменные структурного типа *number int_number*, *number real_number*.

Выходные значения: код результата (1 — больше или равно, 0 — меньше).

int compare_zero(number *one_number);

Описание: функция сравнивает число и 0.

Входные значения: переменная структурного типа *number*.

Выходные значения: код результата (1 — равны, 0 — различны).

void change_zero(number *one_number);

Описание: функция меняет позицию нуля.

Входные значения: переменная структурного типа *number*.

Выходные значения: переменная структурного типа *number* с измененной позицией нуля.

int subtraction(number *int_number, number *real_number);

Описание: функция вычитает из целого числа *number int_number* действительное *number real_number* и записывает результат в структуру целого *number int_number*.

Входные значения: переменные структурного типа *number int_number*, *number real_number*.

Выходные значения: переменная структурного типа *number int_number* с результатом вычитания, код ошибки.

int divide_numbers(number *int_number, number *real_number, number *result_number);

Описание: делит целое число *number int_number* на действительное *number real_number* и сохраняет результат в *number result_number*.

Входные значения: переменные структурного типа *number int_number*, *number real_number*.

Выходные значения: переменная структурного типа *number result_number* с результатом деления, код ошибки.

void output_number(number real_number);

Описание: функция печатает число из структуры *number real_number*.

Входные значения: переменная структурного типа *number real_number*.

Выходные значения: сообщение с результатом деления в нормализованном виде.

Тесты

№	Случай	Целое число	Действительное число	Вывод
1	Отсутствие знака целого числа	a	-	Первым символом целого числа должен быть знак числа.
2	В мантиссе целого числа недопустимые символы	+12a	-	Мантисса целого числа должна состоять только из цифр.
3	В мантиссе целого числа введены пробелы	- 12	-	Не должно быть пробелов между символами.
4	Отсутствие знака действительного числа	+123	a	Первым символом действительного числа должен быть знак числа.

5	В мантиссе действительного числа недопустимые символы	-123	-1b	Мантисса действительного числа должна состоять только из цифр.
6	В мантиссе действительного числа введены пробелы	+123	-2 .3	Не должно быть пробелов между символами.
7	Не введен знак порядка	+123	-2.3E24	Перед порядком действительного числа должен быть знак порядка.
8	В порядке действительного числа недопустимые символы	+123	-0.02E-a	Порядок действительного числа должен состоять только из цифр.
9	Неверная длина мантиссы целого числа	-99...9 (31 цифра 9)	-	Длина мантиссы целого числа должна быть меньше, либо равна 30 и больше 0.
10	Неверная длина мантиссы действительного числа	-123	+99...9E+1 (31 цифра 9 в мантиссе)	Длина мантиссы действительного числа должна быть меньше, либо равна 30 и больше 0.
11	Неверная длина порядка	-123	+123E+99...9 (6 цифра 9 в порядке)	Длина порядка действительного числа должна быть меньше, либо равна 5 и больше 0.

12	Переполнение порядка	+10000	-1E+99999	Переполнение порядка.
13	Действительное число равно 0	-123	-0	Нельзя делить на 0.
14	Деление 0 на 0	+0	-0	Нельзя делить на 0.
15	Целое число равно 0	+0	-12.3E+1	+0.0E+0
16	Деление одного знака	-144	-1.2E+1	+0.12E+2
17	Деление разных знаков	+144	-1.2E+1	-0.12E+2
18	Округление	+99...9 (30 цифра 9)	+2	+0.5E+30

Выводы

При написании лабораторной работы реализована функция деления целого числа на длинное вещественное, используя алгоритм деления «в столбик». Для ввода чисел использовался массив символов, для хранения, обработки и вывода чисел - переменные структурного типа.

Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Возможный диапазон чисел зависит от выбранного типа, разрядности процессора и памяти выделенной для хранения числа. Для беззнакового целого числа выделяется 64 двоичных разряда, его максимальное значение – $2^{64} - 1$ 8 446 744 073 709 551 615.

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

Точность представления вещественных чисел зависит от размера памяти, выделенной для хранения мантиссы. Для мантиссы числа типа double это значение равно 52 бита, поэтому мантисса числа может иметь значение до 4 503 599 627 370 496.

3. Какие стандартные операции возможны над числами?

Числа возможно складывать, вычитать, умножать, делить, брать остаток от числа и сравнивать числа.

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

Программист может использовать массив чисел или символов, представляя числа в строку, или структуру, поля которой могут быть частями числа.

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

Можно написать собственные функции, реализующие такие операции или использовать функции дополнительных библиотек.