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» (ИУ)
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«ДЛИННАЯ АРИФМЕТИКА»

по курсу:

«ТИПЫ И СТРУКТУРЫ ДАННЫХ»

Вариант: 4

Студент:

Авдейкина Валерия Павловна, группа ИУ7-33Б _____
(подпись, дата)

Руководители:

Преподаватель ИУ7 _____
Силантьева Александра Васильевна (подпись, дата)
Преподаватель ИУ7

Барышникова Марина Юрьевна _____
(подпись, дата)

Проверивший:

Барышникова Марина Юрьевна _____
(подпись, дата)

Оценка: _____

2022 г.

Оглавление

Описание условий задачи

Смоделировать операцию деления целого числа длиной до 30 десятичных цифр на действительное число в форме `±M.NE±K`, где суммарная длина мантиссы `(M+N)` - до 30 значащих цифр, а величина порядка `K` - до 5 цифр. Результат выдать в форме `±0.M1E±K1`, где m1 - до 30 значащих цифр, а `K1` - до 5 цифр.

Техническое задание

1 Входные данные

- Целое число вводится пользователем в виде строки в формате ±Z, где
  - ± – знак числа (+ или -), необязателен
  - Z – десятичные цифры числа, общее количество которых не должно превышать 30
- Действительное число вводится пользователем в виде строки в формате ±M.NE±K, где
  - ±M.N – мантисса числа (разделитель – обязательно точка):
    - ± – знак мантиссы (+ или -), необязателен, не может повторяться
    - M, N – десятичные цифры мантиссы, общее количество которых не должно превышать 30, допускается вариант с отсутствием M или N
  - E±K – порядок числа (необязателен):
    - E – латинская, допускается как строчная E, так и прописная e
    - ± (2) – знак порядка (+ или -), необязателен, не может повторяться
    - K – десятичные цифры порядка, общее количество которых не должно превышать 5

2 Выходные данные

- Действительное число выводится на экран в нормальном виде, в строки в формате ±0.M1E±K1, где
  - ±0.M1 – мантисса числа
    - M – десятичные цифры мантиссы, количество которых не превышает 30, лишние нули в конце числа выводиться не будут
  - E±K1 – порядок числа:

- `K1` – десятичные цифры порядка, общее количество которых не превышает 5

3 Задача, реализуемая программой

Программа реализует деление введенных длинных чисел – целого на действительное – с помощью методов длинной арифметики и выводит результат вычислений на экран.

4 Способ обращения к программе

Программа вызывается в командной строке без каких-либо аргументов. Ввод данных производится с клавиатуры после соответствующего приглашения к вводу («`Введите...`»).

5 Возможные аварийные ситуации и ошибки со стороны пользователя

1. Переполнение (выход результата за границы допустимых значений)
2. Деление на нуль
3. Некорректный ввод целого или действительного:
    - 3.1. Пустой ввод
    - 3.2. Повторение знака
    - 3.3. Недопустимые символы
    - 3.4. Превышение допустимого количества цифр
    - 3.5. Переполнение буфера в целом *(поведение программы неопределено)*

Описание внутренних структур данных

В ходе работы были составлены две структуры данных:

1. Структура `long_float` для представления длинного действительного числа, содержащая следующие поля:
   - `mantissa` – массив из `LEN_MANTISSA` элементов типа `digit_t` (`uin8_t`), хранящий цифры мантиссы числа (под нулевым индексом хранится самый старший разряд)
   - `exponent` – переменная типа `int`, хранящая значение порядка числа с учетом знака
   - `mant_sign` – переменная типа `bool` (`uint8_t`), хранящая информацию о том, имеет ли мантисса знак «минус»
   - `is_null` – переменная типа `bool` (`uint8_t`), хранящая информацию о том, является ли число нулем/машинным нулем
   - `is_inf` – переменная типа `bool` (`uint8_t`), хранящая информацию о том, выходит ли число за границы допустимых значений

```c
typedef uint8_t bool;

typedef uint8_t digit_t;

typedef struct long_float
{
  digit_t mantissa[LEN_MANTISSA];
  int exponent;
  bool mant_sign;
  bool is_null;
  bool is_inf;
} long_float_t;

// #define LEN_MANTISSA 30 + 2
```

Листинг 1. Структура для представления длинного действительного числа

2. Целое число вводится в виде строки, обрабатывается и переводится в формат длинного действительного; хранится в описанной структуре `long_float`

3. Структура `str_float` для представления информации о входной строке, предположительно содержащей вещественное число, содержащая следующие поля:
    - `before_dot` – переменная типа `size_t`, хранящая количество цифр, встреченных в строке до точки
    - `after_dot` – переменная типа `size_t`, хранящая количество цифр, встреченных в строке после точки
    - `exp_len` – переменная типа `size_t`, хранящая количество цифр порядка числа
    - `exp_sign` – переменная типа `bool`, хранящая информацию о том, был ли встречен знак порядка числа
    - `mant_sign` – переменная типа `bool`, хранящая информацию о том, был ли встречен знак мантиссы числа
    - `has_dot` – переменная типа `bool`, хранящая информацию о том, была ли встречена точка в качестве разделителя частей числа

```
typedef struct str_float
{
    size_t before_dot;
    size_t after_dot;
    size_t exp_len;
    bool exp_sign;
    bool mant_sign;
    bool has_dot;
} str_float_t;
```

Листинг 2. Структура для представления информации о входной строке

Итог (занимаемая память):

- Структура `long_float` имеет размер 39 байт
- Структура `str_float` имеет размер 15 байт

Алгоритм

Выбранный для выполнения задачи алгоритм заключается в приведении целого и действительного длинных чисел к нормализованному виду `±0.M1E±K1` и `±0.M2E±K2`. Тогда поля результата деления мы будем получать следующим образом:

1. Порядок результата = K1 - K2
2. Знак результата = знак первого XOR знак второго
3. Мантиссу результата будем искать, представив операцию деления мантисс двух чисел делением «в столбик» – сведем ее к операции вычитания со сдвигами (пока не заполним всю мантиссу результата и мантисса делимого не нулевая):
    3.1. Вычитание одной мантиссы из другой является поэлементным вычитанием одного массива из другого, начиная с последнего элемента
    3.2. «Занятие» делимым новой цифры справа является сдвигом элементов делителя вправо на один элемент
    3.3. Нахождение очередной цифры результата является вычитанием делителя из делимого, пока делимое больше делителя (*Примечание:* мантисса «больше»/«меньше» другой мантиссы, если при одном индексе отличный элемент первой мантиссы «больше»/«меньше» элемента с этим же индексом второй мантиссы)
    3.4. Если до начала деления нормализованная мантисса делимого больше нормализованной мантиссы делителя, порядок увеличивается на единицу
    3.5. Если при поиске очередной цифры результата делимое меньше делителя, «занимаем» новую цифру
    3.6. В конце каждого шага деления нормализуем мантиссу делимого

В реализации решения задачи были использованы следующие основные функции:

Модуль	Функция
longio	`int input_float(long_float_t *const number)`
	`void output_float(long_float_t number)`
	`int input_int(long_float_t *const number)`
	`size_t normalize_mant(digit_t *const mant)`
	`void shift_mant(digit_t *const mant, const bool dir, const size_t offset)`
	`void copy_mant(digit_t *const dest, digit_t *const src)`
	`bool is_null_mant(digit_t *const mant)`
	`void round_number(long_float_t *const num)`
	`void init_number(long_float_t *const n)`
longproc	`int divide_long_float(long_float_t numerator, long_float_t divisor, long_float_t *const res)`
	`int compare_mant(digit_t *const mant1, digit_t *const mant2)`
	`void diminute_mant(digit_t *const mant1, digit_t *const mant2)`

Таблица 1. Модули и основные функции, используемые в алгоритме

Функция	Описание
`input_float`	Принимает: указатель на переменную структурного типа `long_float_t` Выполняет: ввод строки с вещественным числом; проверку на соответствие строки виду вещественного числа; перевод полученной строки в переменную структурного типа. Возвращает: код ошибки/успеха
`output_float`	Принимает: переменную структурного типа `long_float_t` Выполняет: вывод действительного числа на экран Возвращает: –
`input_int`	Принимает: указатель на переменную структурного типа `long_float_t` Выполняет: ввод строки с целым числом; проверку на соответствие строки виду целого числа; перевод полученной строки в переменную структурного типа Возвращает: код ошибки/успеха
`normalize_mant`	Принимает: массив типа `digit_t` Выполняет: сдвиг элементов массива влево, избавляясь от ведущих нулей. Возвращает: количество удаленных нулей
`shift_mant`	Принимает: массив типа `digit_t`, направление (RIGHT или LEFT), размер сдвига Выполняет: сдвиг элеметов массива в указанном направлении на указанную длину сдвига. Возвращает: –
`copy_mant`	Принимает: (куда) массив типа `digit_t`, (откуда) массив типа `digit_t` Выполняет: последовательное копирование элементов из первого массива во второй Возвращает: –
`is_null_mant`	Принимает: массив типа `digit_t` Выполняет: проверку массива на содержание лишь нулевых элементов Возвращает: TRUE или FALSE
`round_number`	Принимает: указатель на переменную структурного типа `long_float_t` Выполняет: округление числа с расстановкой флагов нуля и переполнения. Возвращает: –
`init_number`	Принимает: указатель на переменную структурного типа `long_float_t` Выполняет: инициализацию длинного числа (обнуление всех полей структуры). Возвращает: –

`divide_long_float`	Принимает: (структурный тип `long_float_t`) делимое, делитель, указатель на переменную для результата Выполняет: деление делимого на делитель; анализ результата; помещение результата по указателю Возвращает: код ошибки/успеха
`compare_mant`	Принимает: два массива типа `digit_t` Выполняет: сравнение мантисс, хранящихся в массивах поразрядного (до первого несовпадающего элемента при наличии, иначе до конца) Коды: EQUAL/FIRST_LARGER/SECOND_LARGER
`diminute_mant`	Принимает: (массивы типа `digit_t`) уменьшаемое, вычитаемое Выполняет: поразрядное вычитание из элементов уменьшаемого элементы вычитаемого, начиная с последней, учитывая переполнение цифр. Возвращает: –

Таблица 2. Описание основных функций

Тестирование

Запуск функциональных тестов осуществляется с помощью вызова скрипта `./func_tests.sh` из папки `./func_tests/scripts`.

- Позитивные тесты:

№	Описание	Входные данные	Результат
1	Деление нуля	0 1	0
2	Деление на очень большое	1 999e99999	0
3	Деление с округлением	8 3	+0.2(28 «6»)7E+1
4	Деление отрицательного на отрицательное	-1 -1	+0.1E+1
5	Деление положительного на отрицательное	1 -1	-0.1E+1
6	Порядок делителя отрицательный	1 1e-1	+0.1E+2
7	Порядок делителя положителен	1 1e1	+0.1E+0
8	Вещественное число написано в неполной форме	1 1.	+0.1E+1
9	Деление без округления	9 2	+0.45E+1
10	Макисмум цифр в целом на 1	30 * «9» 1	+0.(30 девяток)E+30
11	Максимум цифр в целом и максимум цифр мантиссы в действительном	30 * «9» 30 * «9»	+0.1E+1
12	Деление с цепочным округлением	30 * «9» 2	+0.5E+30

- Негативные тесты:

№	Описание	Входные данные	Сообщение/код выхода
1	Посторонние символы при вводе целого числа	abc 1e1	Ошибка: Неверный ввод…/ERR_FORMAT

2	Посторонние символы при вводе вещественного числа	1 1abc1	Ошибка: Неверный ввод…/ERR_FORMAT
3	Превышение количества допустимых цифр (целое число)	тридцать пять единиц 1e1	Ошибка: Неверный ввод…/ERR_FORMAT
4	Превышение количества допустимых цифр (мантисса действительного)	1 (31 «1»)e1	Ошибка: Неверный ввод…/ERR_FORMAT
5	Превышение количества допустимых цифр (порядок действительного)	1 1e123456	Ошибка: Неверный ввод…/ERR_FORMAT
6	Дублирование знака (целое число)	++1 1e1	Ошибка: Неверный ввод…/ERR_FORMAT
7	Дублирование знака (действ. число)	1 ++-1e1	Ошибка: Неверный ввод…/ERR_FORMAT
8	Деление на нуль	1 0	Ошибка: Деление на нуль... /ERR_ZERODIV
9	Деление на очень маленькое	1 0.001e-99999	Ошибка: Переполнение.../ERR_INF

Контрольные вопросы

1 Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел напрямую зависит от области памяти, выделяемой под переменную, знака числа (наличия или отсутствия), а также типа представления числа (вещественное или целое). К примеру, если целая беззнаковая переменная типа integer занимает 2 байта (16 бит), то диапазон её значений: `0..2^16-1 = 0..65535`. Если же она может принимать также и положительные значения, то диапазон: `-2^15..2^15-1`

2 Какова возможная точность представления чисел?

Вещественные числа хранятся в представлении с плавающей точкой в виде `X = M * 10^K`, где `M` – мантисса, представленная правильной дробью (в интервале [0.1..1)). Точность представленя вещественного числа зависит от максимально возможной длины мантиссы, зависящей от размера выделяемой памяти и знака числа (наличия или отсутствия); если длина мантиссы выходит за границы разрядной сетки, то происходит округление.

3 Какие стандартные операции возможны над числами?

Стандартные операции над числами – сложение, вычитание, умножение, деление, сравнение чисел.

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

Для представления чисел, превышающих возможный диапазон, может быть использован массив из цифр (например, переменных занимающих 1 байт) – простейший случай представления целого беззнакового числа.

В случае с небольшим порядком (до 5 знаков) можно использовать, например, структуру, содержащую массив цифр мантиссы, порядок с учетом знака (если он «помещается» в простые типы) и флаг для хранения знака мантиссы.

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

Операции над большими числами можно выполнять путём последовательного выполнения операций над всеми цифрами, составляющими число, начиная с конца массива .

Выводы

В ходе работы было выяснено, что хранение длинных чисел в виде определенных структур данных и использование длинной арифметики позволяют работать с длинными числами быстро и эффективно, однако алгоритмы вычислений становятся сложнее.