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 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

“Обработка больших чисел”

Студент **Зернов Георгий Павлович**

Группа **ИУ7-34Б**

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

Студент _____ **Зернов Г.П.**

Преподаватель _____ **Никульшина Т.А.**

2024

Оглавление

Условие задачи

Смоделировать операцию деления целого числа длиной до 40 десятичных цифр на действительное число в форме ±m.n E K, где суммарная длина мантиссы (m+n) - до 40 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме ±0.m1 E ±K1, где m1 - до 40 значащих цифр, а K1 - до 5 цифр.

Техническое задание

Реализуемая в программе задача

Цель работы: реализация деления целого числа на действительное число, в случае если числа выходят за разрядную сетку локальной машины.

Входные данные

Делимое и делитель, вводимые последовательно в различных строках, без пробелов:

- Делитель — целое число, соответствующее следующей нотации: ^[+-]?[0-9]+$. Максимально возможное число цифр - 40.

- Делимое — целое число, соответствующее следующей нотации: ^[-+]?(([0-9]+([.][0-9]+)?)|([.][0-9]+)|([0-9]+[.]))([eE][-+]?[0-9]+)?$, т.е. число вида ±m.n E K, где длина мантиссы (m + n) не более 40
цифр и экспонента K не более 5 цифр.

Ввод делимого и делителя происходит последовательно, после приглашений к вводу. Во входных строках должны отсутствовать любые символы, кроме символа перевод строки в конце, не входящие в нотацию.

Выходные данные

Результат деления — вещественное число в нормализованной форме, т.е. число вида ±0.m1 E K1, где длина мантиссы m1 — не более 40 знаков и длина экспоненты K — не более 5 знаков.

Способ обращения к программе

Чтобы обратиться к программе, пользователю необходимо запустить файл app.exe.

Возможные ошибки пользователя

1. Невозможность интерпретировать ввод пользователя как валидную строку, введено более 50 символов и/или последним символом не является символ перевода строки – функция возвращает ошибку INVALID_STRING_ERROR.
2. Введённая в качестве целого числа строка не соответствует нотации – функция возвращает ошибку NOT_AN_INT_ERROR.
3. Введённое целое число содержит более 40 цифр – функция возвращает EXCEEDED_MAX_DIGITS_NUM_ERROR.
4. Введённая в качестве вещественного числа строка не соответствует нотации – функция возвращает ошибку NOT_A_FLOAT_ERROR.
5. Введённое вещественное число содержит более 40 цифр в мантиссе – функция возвращает EXCEEDED_MAX_MANTISSA_LEN_ERROR.
6. Введённое вещественное число содержит более 5 цифр в экспоненте – функция возвращает INVALID_EXPONENT_ERROR.

Внутренние структуры данных

Структура, представляющее длинное целое число.

- int sign – хранит знак числа, значение может быть 1, если число положительное или нуль, или -1 если оно отрицательное.
- int digits[MAX_DIGITS_NUM] – хранит цифры числа в обратном порядке, в количестве не более MAX_DIGITS_NUM (40).

Листинг структуры:

```
typedef struct
{
    int sign;
    int digits[MAX_DIGITS_NUM];
} big_int_t;
```

Структура, представляющее длинное вещественное число.

- int sign – хранит знак числа, значение может быть 1, если число положительное или нуль, или -1 если оно отрицательное.
- int mantissa[MAX_MANTISSA_LEN] – хранит цифры мантиссы числа в обратном порядке, в количестве не более MAX_MANTISSA_LEN (41).
- int exponent – хранит экспоненту числа.

Листинг структуры:

```
typedef struct
{
    int sign;
    int mantissa[MAX_MANTISSA_LEN];
    int exponent;
} big_float_t;
```

Алгоритм программы

1. Считывается строка, при превышении максимальной длины или если строка не оканчивается символом перевода строки выводится сообщение об ошибке.
2. Строка проверяется на соответствие нотации валидного целого числа, в случае соответствия приводится к типу длинного целого числа, иначе выводится соответствующее сообщение об ошибке.
3. Считывается строка, при превышении максимальной длины или если строка не оканчивается символом перевода строки выводится сообщение об ошибке.

4. Строка проверяется на соответствие нотации валидного вещественного числа, в случае соответствия приводится к типу длинного вещественного числа, иначе выводится соответствующее сообщение об ошибке.

5. Считанное длинное целое число приводится к типу длинного вещественного числа.

6. Второе число проверяется на равенство 0, в случае равенства выводится соответствующее сообщение и выполнение прекращается.

7. Создаётся длинная вещественная переменная, в которую будет записан результат деления первого числа на второе.

8. Изменяя экспоненту, массивы цифр делителя и делимого сдвигают в сторону больших разрядов так, чтобы делитель был не более чем в 10 раз больше делителя.

9. Производится деление в столбик, результат записывается в ранее созданную переменную.

10. Если в результате деления произошло переполнение экспоненты, выводится соответствующее сообщение.

11. Результат деления выводится в нормализованном виде.

Функции программы

- int is_big_int(char *str) – возвращает EXIT_SUCCESS, если переданная в неё строка является валидным целым числом, NOT_AN_INT – если строка не соответствует нотации числа, и EXCEEDED_MAX_DIGITS_NUM – если в числе более 40 цифр.
- void to_big_int(char *str, big_int_t *num) – приводит переданную строку к пользовательскому типу big_int_t. Работает только для строк, от которых функция is_big_int вернёт EXIT_SUCCESS.
- int is_big_float(char *str) – возвращает EXIT_SUCCESS, если переданная в неё строка является валидным вещественным числом, NOT_A_FLOAT – если строка не соответствует нотации числа, INVALID_EXPONENT – если

экспонента числа содержит более 5 цифр и EXCEED-ED_MAX_MANTISSA_LEN – если в мантиссе числа более 40 цифр.

- void to_big_float(char *str, big_float_t *num) – приводит переданную строку к пользовательскому типу big_float_t. Работает только для строк, от которых функция is_big_float вернёт EXIT_SUCCESS.
- void big_int_to_big_float(big_int_t *int_num, big_float_t *float_num) – приводит число типа bog_int_t к типу big_float_t.
- void shift_arr(int *arr, size_t arr_size, int shift) – сдвигает с заполнением 0 элементы массива на shift позиций вправо,
- int cmp_arr(int *a, int *b, size_t size) – сравнивает два массива целых чисел одинаковой длины между собой по правилам сравнения строк, начиная с конца.
- void sub_arr(int *minuend, int *subtrahend, size_t arr_size) – вычитает из массива целых чисел minuend массив subtrahend по правилам вычитания целых чисел, считая что старшие разряды находятся в конце массива.
- big_float_t division(big_float_t *dividend, big_float_t *divisor) – возвращает big_float_t – результат деления big_float_t на big_float_t, при условии что делитель отличен от 0.
- void round_big_float(big_float_t *num, size_t round_digit) – округляет число по заданному разряду.
- void printf_normalized(big_float_t *num) – выводит на экран число типа big_float_t в нормализованном виде.
- int main(void) – исполняет основной алгоритм программы.

Тесты

Первое число	**Второе число**	**Код возврата**	**Тестовый случай**	**Ожидаемый результат**
Пустая строка		INVALID_STRI	Пустая строка	Ошибка

		NG_ERROR		считывания строки
+1	Строка длиной больше чем MAX_BUFF_LEN	INVALID_STRING_ERROR	Строка, не завершающаяся “\n”	Ошибка считывания строки
Строка		NOT_AN_INT_ERROR	Первая строка не соответствует нотации целого числа	Ошибка валидации первой строки
123	3[55 цифр]	EXCEEDED_MAX_DIGITS_NUM_ERROR	Вылет за границы ввода строки	Ошибка валидации первой строки
-7	+1a2	NOT_A_FLOAT_ERROR	Вторая строка не соответствует нотации вещественного числа	Ошибка валидации второй строки
42	+[41 цифра]E1	EXCEEDED_MAX_MANTISSA_LEN_ERROR	Число содержит более 40 цифр в мантиссе	Ошибка валидации второй строки
123	1E1000000	INVALID_EXPONENT_ERROR	Экспонента содержит больше 5 цифр	Ошибка валидации второй строки
+1	0	EXIT_SUCCESS	Деление на 0	Сообщение о делении на 0
0	100	EXIT_SUCCESS	Деление 0 на число	+0.0e0
-99999999	1e0	EXIT_SUCCESS	Деление на 1	-0.99999999e8
100	3	EXIT_SUCCESS	Деление с отсутствием	0.[3x40]e+2

			округленя	
11111	271.0	EXIT_SUCCESS	Целочисленное деление	0.41e+2
23498	234.98e2	EXIT_SUCCESS	Одинаковые числа	+0.1e+1
1	0.6	EXIT_SUCCESS	Деление с округлением	+0.1[6x38]7e+1
[9x40]	[9x40]	EXIT_SUCCESS	Максимальные размеры мантисс	+0.1e+1
325	+5e99999	EXIT_SUCCESS	Делитель имеест масимальную экспоненту	+0.65e-99997
1000	1000e-99999	EXIT_SUCCESS	Переполнение экспоненты в положительную сторону	Сообщение о переполнении экспоненты
45	-9000e+99999	EXIT_SUCCESS	Переполнение экспоненты в отрицательную сторону	Сообщение о переполнении экспоненты
1	[9x40]E99961	EXIT_SUCCESS	Переполнение экспоненты после округления	Сообщение о переполнении экспоненты
[9x40]	2	EXIT_SUCCESS	Циклическое округление	+0.5e+40
12345	+3.e1	EXIT_SUCCESS	Оба числа положительные	+0.4115e+3
-440	-13.75e98	EXIT_SUCCESS	Оба числа отрицательные	+0.32e-96

10	-90.	EXIT_SUCCESS	Первое число положительное, второе отрицательное	-0.[1x40]e+1
-179	.179e2	EXIT_SUCCESS	Первое число отрицательное, второе положительное	-0.1e+2

Вывод

Для обработки чисел, выходящих за разрядную сетку машины можно хранить цифры таких чисел в массиве символов и операции над ними проводить поразрядно.

Контрольные вопросы:

Каков возможный диапазон чисел, представляемых в ПК?

Зависит от разрядности процессора ПК: например, в 64-разрядном процессоре числа могут принимать значения от -2^{63} до $2^{63}-1$ (от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807).

Какова возможная точность представления чисел, чем она определяется?

Точность представления числа задаётся размером мантиссы. Согласно стандарту IEEE754 для чисел двойной точности она 52 бинарных разряда и 23 разряда для чисел с одинарной точностью. Соответствующие значения в десятичной СС: 2^{52} = 4503599627370496 и 2^{23} = 8388608.

Какие стандартные операции возможны над числами?

Сравнение, сложение, вычитание, умножение, деление.

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

Для хранения цифр числа программист может выбрать массив и хранить в нём число поразрядно.

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

Для осуществления операций над числами, выходящими за рамки машинного представления, требуется разбить операцию на операции над числами помещающимися в машинное представление т.е., например выполнять операцию поразрядно.