



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 ДЛИННАЯ АРИФМЕТИКА

Студент Жаринов М. А.

Вариант 2

Группа ИУ7-32Б

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

Студент _____ Жаринов М. А.

Преподаватель _____ Барышникова М. Ю.

2024

Описание условия задачи

Смоделировать операцию умножения действительного числа в форме $\pm m.n \text{ E } \pm K$, где суммарная длина мантиссы ($m+n$) - до 40 значащих цифр, а величина порядка K - до 5 цифр, на целое число длиной до 30 десятичных цифр. Результат выдать в форме $\pm 0.m1 \text{ E } \pm K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр.

Техническое задание

Исходные данные:

Вещественное число: Строка, содержащая число в формате

$[+ -]?m.?n(E[+ -]?K)?$, где $(m+n)$ до 40 значащих цифр, K по модулю не превышает 99999. Если не указан знак, то знак по умолчанию плюс.

Целое число: Строка, содержащая число в формате $[+ -]?m1$, где m до 30 значащих цифр. Если не указан знак, то знак по умолчанию плюс.

Выходные данные:

Вещественное число в формате $\pm 0.m1 E \pm K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр.

Реализуемая задача:

Умножение вещественного числа на целое

Способ обращения к программе:

Строка запуска программы ./app.exe

После запуска вводятся по очереди вещественное и целое число. Признаком окончания ввода – Enter

Аварийные ситуации:

1. Пустой ввод. Код ошибки 1 (ERR_EMPTY_INPUT)
2. Слишком большой ввод. Код ошибки 2 (ERR_INPUT_TOO_LONG)
3. Неверный формат числа. Код ошибки 3 (ERR_INPUT_INVALID)
4. Получение машинного нуля. Код ошибки 4 (WARN_ZERO)
5. Получение машинной бесконечности. Код ошибки 5 (WARN_INF)

Внутренние структуры данных

Вещественные числа хранятся в структуре `real_t`.

```
typedef struct
{
    int8_t mnt_sign;
    uint8_t mantissa[MANTISSA_LEN];
    int32_t exponent;
} real_t;
```

Поле `mnt_sign` – хранит -1 или 1 в зависимости от знака мантиссы

Поле `mantissa` – массив байтов для хранения мантиссы (одна цифра на байт)

Поле `exponent` – порядок вещественного числа

Число всегда хранится в нормализованном виде (первый элемент массива цифр – наибольшая значащая цифра)

Целые числа хранятся в структуре `int_t`.

```
typedef struct
{
    int8_t sign;
    uint8_t digits[INT_LEN];
} int_t;
```

Поле `sign` – хранит -1 или 1 в зависимости от знака

Поле `digits` – массив байтов для хранения числа (одна цифра на байт)

Число всегда хранится в следующем виде : последний элемент массива цифр – разряд единиц

Описание алгоритма

1. Программа запрашивает вещественное число, считывает его и записывает в структуру в нормализованном виде
2. Программа запрашивает целое число, считывает его и записывает в структуру в нормализованном виде
3. Программа выполняет умножение первого числа на второе, округляет при необходимости до 30 знаков и проверяет переполнение или получение машинного нуля
4. Программа выводит результат на экран

Основные функции:

Функция для умножения вещественного числа на целое (изменяет и принимает структуру вещественного числа по указателю, принимает структуру целого по указателю), возвращает код ошибки

```
int multiply_real_by_int(real_t *real_num, int_t *int_num);
```

Функция обработки кода ошибки и печати соответствующего сообщения.

Принимает и возвращает код ошибки

```
int process_error(int err_code);
```

Функция чтения целого числа из потока ввода. Возвращает код ошибки, изменяет структуру по указателю

```
int read_int(int_t *int_num);
```

Функция чтения вещественного числа из потока ввода. Возвращает код ошибки, изменяет структуру по указателю

```
int read_real(real_t *real_num);
```

Функция печати приветствия для целого числа. Принимает длину линейки

```
int read_real(real_t *real_num);
```

Функция печати приветствия для вещественного числа. Принимает длину линейки

```
int read_real(real_t *real_num);
```

Функция печати вещественного числа. Принимает структуру

вещественного числа по указателю

```
int read_real(real_t *real_num);
```

Набор тестов

Описание теста	Вещественное число	Целое число	Вывод
Умножение положительного вещественного числа на положительное целое	22E3	3	+0.66E+5
Умножение отрицательного вещественного числа на положительное целое	-22E3	3	-0.66E+5
Умножение положительного вещественного числа на отрицательное целое	11E4	-45	-0.495E+7
Умножение отрицательного вещественного числа на отрицательное целое	-11E4	-45	+0.495E+7
Умножение целого числа на целое	13	16	+0.208E+3
Умножение	+99999999999999	1	+0.1E+99999

максимально возможного целого числа на 1	99999999999999 9999999999999E 99958		
Получение максимального возможного числа	0.11111111111111 11111111111111 1E99999	9	+0.9999999999 999999999999 99999999999E +99999
Умножение числа с минимально возможной экспонентой и мантиссой меньше 1 на максимальное	0.001E-99999	999999..999	+0.9999999999 999999999999 99999999999E -99972
Округление результата с повышением порядка	83333..3333E0	12	+0.1E+41
Умножение максимальных мантисс	99999999999999 99999999999999 99999999999999	99999999999999 9999999999	+0.9999999999 999999999999 99999999999E +70
Умножение на 1 числа, некорректного при нормализации	10E99999	1	Машинная бесконечност ь!
Вещественное число с слишком большой экспонентой	10E100000	-	Неверный ввод!
Вещественное число с слишком большой	+19999999999999 99999999999999	-	Неверный ввод!

мантиссой	9999999999999999 E99		
Вещественное число с маленькой буквой e	10e10	-	Неверный ввод!
Слишком большое целое число	1	1000000000000000 0000000000000000 00001	Слишком большой ввод!
Вещественное число вместо целого	12.1	12.0	Неверный ввод!
Получение машинного нуля	0.0001E-99999	1	Машинный ноль!
Умножение на 0	12	0	+0.E+0
Получение слишком большого числа	0.1E99998	1000	Машинная бесконечност ь!
Пустой ввод			Пустой ввод!

Ответы на контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Для 64-х разрядного процессора можно хранить числа от 0 до 18 446 744 073 709 551 615 ($2^{64}-1$) или числа со знаком от -2^{63} до $2^{63}-1$ для целых чисел.

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

Точность представления вещественных чисел определяется количеством памяти, выделенной под хранение мантиссы. В языке Си, при использовании типа `double`, под мантиссу выделено 52 бита, то есть максимальное значение мантиссы может составлять 4 503 599 627 370 496 (2^{52}), примерно 10 десятичных разрядов.

3. Какие стандартные операции возможны над числами?

Стандартные операции над числами в Си: сложение, вычитание, умножение, деление, взятие остатка, операции сравнения, присвоения, инкремент, декремент

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

В таком случае программист может использовать структуру, содержащую цифры числа в массиве и другую нужную информацию

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

Для осуществления таких операций необходимо реализовать собственные типы данных и функции для этого действия (как в данной лабораторной работе), а сами вычисления производить в столбик

Вывод

При необходимости проводить операции над числами, которые слишком велики для стандартных типов в языке программирования, программист должен самостоятельно реализовать необходимые структуры данных и функции для работы с ними. В данной работе реализованы типы данных для целых и вещественных чисел с точностью и размером, превышающей ограничения стандартных типов языка Си, а также функция их умножения. Такие типы данных могут быть необходимы для выполнения физических или финансовых расчетов, где необходима крайне высокая точность и/или большой размер чисел.