



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 **«ДЛИННАЯ АРИФМЕТИКА» (ВАРИАНТ 2)**

Студент Равашдех Фадей Хешамович

Группа ИУ7 – 35Б

Преподаватель Силантьева А. В.

Оглавление

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	<u>2</u>
<u>ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ.....</u>	<u>5</u>
<u>ОПИСАНИЕ ФУНКЦИЙ.....</u>	<u>7</u>
<u>ОПИСАНИЕ АЛГОРИТМА.....</u>	<u>11</u>
<u>НАБОР ТЕСТОВ.....</u>	<u>12</u>
<u>ОЦЕНКА ЭФФЕКТИВНОСТИ.....</u>	<u>13</u>
<u>ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	<u>14</u>
<u>ВЫВОД.....</u>	<u>15</u>

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Смоделировать операцию умножения действительного числа в форме $\pm m.n E \pm K$, 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 $\pm 0.m_1 E \pm K_1$, где m_1 - до 30 значащих цифр, а K_1 - до 5 цифр.

Описание технического задания

Входные данные:

Действительное число: строка, содержащая действительное число в виде $\langle (\pm)\backslash m.n(E\backslash e)(\pm)\backslash K \rangle$. Если не указать знак перед числом и/или экспонентой, то по умолчанию будет считаться за '+'. Если не указать экспоненту, то по умолчанию будет считаться за 'E+0'. Суммарная длина $\langle m+n \rangle$ — 1-40 цифры; длина порядка — 1-5 цифр.

Целое число: строка, содержащая целое число в виде $\langle \pm\backslash m \rangle$. Если не указать знак перед числом, то по умолчанию будет считаться за '+'. Длина модуля числа $\langle m \rangle$ — 1-30 цифр.

Выходные данные:

Длинное число в виде $\langle \pm\backslash 0.m_1 E \pm\backslash K_1 \rangle$. Длина мантиссы $\langle m_1 \rangle$ — до 30 цифр; длина порядка $\langle K_1 \rangle$ — до 5 цифр.

Задача программы:

Умножение действительное числа на целое.

Обращение к программе:

Запускается через терминал из папки с программой:

```
$ ./app.exe
```

Выбирается пункт в меню:

- «1. Вывод информации о программе.
- 2. Умножение вещественного числа на целое.
- 3. Выход из программы»

При выборе пункта 2 (умножение) вводится сначала действительное число, потом целое.

Аварийные ситуации и ошибки пользователя:

1. Выбран некорректный пункт меню.

Код возврата – 100

2. Ошибка при вводе действительного числа

Некорректный ввод.

Код возврата – 1.

3. Ошибка при вводе целого числа

Некорректный ввод.

Код возврата – 2.

4. Ошибка при вводе целого числа.

Некорректный ввод: строка с действительным числом не подходит под регулярное выражение-маску "`^([+-]?([0-9]+([.][0-9]*)?([eE][+-]?[0-9]+)?|([.][0-9]+([eE][+-]?[0-9]+)?))$`"

Код возврата – 3.

5. Некорректный ввод: строка с целым числом не подходит под регулярное выражение-маску `"^([+-]?[0-9]+)$"`
Код возврата – 4.
6. Некорректный ввод/переполнение в вычислениях: превышение длины порядка (больше 5 символов).
Код возврата – 5.
7. Превышение длины мантиссы действительного числа.
Код возврата – 6.
8. Превышение длины целого числа.
Код возврата – 7.
9. Превышение длины действительного числа.
Код возврата — 8.
10. Ввод пустой строки.
Код возврата — 9.
11. При умножении или округлении получился нуль.
Код возврата — 10.
12. Ошибка при вводе пункта. EOF
Код возврата — 11.

Во всех аварийных случаях в консоль выводится сообщение, поясняющее из-за чего произошла ошибка.

ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ

После ввода строки с числом, оно переводится в соответствующую структуру. Всего Существует три структуры: для действительного, для целого и для считаемого.

Действительное число обрабатывается и записывается в структуру `real_num`.

Структура `real_num`:

```
#define MANTISS_SIZE 40
typedef struct
{
    int8_t mantiss_sign;
    int8_t mantiss_digits[MANTISS_SIZE];
    int exp;
} real_num;
```

Поля структуры:

`mantiss_sign` – знак мантиссы;

`mantiss_digits` — цифры мантиссы;

`exp` — значение экспоненты (порядка).

Целое число обрабатывается и записывается в структуру `int_num`.

Структура `int_num`:

```
#define INT_SIZE 30

typedef struct
{
    int8_t sign;
    int8_t digits[INT_SIZE];
} int_num;
```

Поля структуры:

`sign` – знак числа;

`digits` — цифры числа.

Считываемое число записывается в структуру `calc_num`.

Структура `calc_num`:

```
#define CALC_MANTISS_SIZE MANTISS_SIZE + INT_SIZE

typedef struct
{
    int8_t mantiss_sign;
    int8_t mantiss_digits[CALC_MANTISS_SIZE];
    int exp;
} calc_num;
```

Поля структуры:

`mantiss_sign` – знак мантиссы;

`mantiss_digits` — цифры мантиссы;

`exp` — значение экспоненты (порядка).

ОПИСАНИЕ ФУНКЦИЙ

Основные функции:

```
/**
 * @brief Управление умножением
 *
 * @return int код выхода
 */
int multiplication_controller(void);
```

```
/**
 * @brief Основная функция программы
 *
 * @return int код выхода
 */
int main(void);
```

Функции ввода/вывода:

```
/**
 * @brief Очищение буфера ввода
 *
 */
void clear_buff(void);
```

```
/**
 * @brief Выводит меню и запрашивает действие
 *
 * @param option выбранный пункт
 */
void menu_get_selected_option(int *option);
```

```
/**
 * @brief Выводит информацию о программе
 */
void info(void);
```

```
/**
 * @brief Ввод действительного числа
 *
 * @param real_num_s строка с действительным числом
 * @return int код возврата
 */
int input_real_num(char *real_num_s);
```

```
/**
 * @brief Ввод целого числа
 *
 * @param integer_num_s строка с целым числом
 * @return int код возврата
 */
int input_int_num(char *integer_num_s);
```



```

/**
 * @brief Вывод результата
 *
 * @param result_num строка с результатом
 */
void print_result(char *result_num);

/**
 * @brief Перевод из строки в структуру действительного числа
 *
 * @param real_num_str строка с действительным числом
 * @param real_num действительное число
 * @return int код возврата
 */
int str_to_real_num(char *real_num_str, real_num *real_num);

/**
 * @brief Перевод из строки в структуру целого числа
 *
 * @param integer_num_str строка с целым числом
 * @param integer_num целое число
 * @return int код возврата
 */
int str_to_int_num(char *integer_num_str, int_num *integer_num);

/**
 * @brief Перевод структуры результата в строку
 *
 * @param res_num структура результата
 * @param res_num_str строка результата
 * @return int код возврата
 */
int res_to_str(calc_num res_num, char *res_num_str);

/**
 * @brief Произведение чисел
 *
 * @param real_num_str строка с действительным числом
 * @param integer_num_str строка с целым числом
 * @return int код возврата
 */
int calculate(real_num real_num, int_num integer_num, calc_num
*res_num);

```

Функции основной логики программы:

```

/**
 * @brief Сдвиг цифр мантиисы влево
 *
 * @param arr массив цифр мантиисы
 * @param n длина массива
 * @param shift сдвиг
 */

```

```

void shift_to_left(int8_t *arr, size_t n, size_t shift);
/**
 * @brief Подсчет конечных нулей
 *
 * @param arr массив цифр мантиисы
 * @param n длина массива
 * @return int количество нулей после числа
 */
int count_zeros(int8_t *arr, size_t n);

/**
 * @brief Подсчет значащих цифр в мантиисе
 *
 * @param arr массив цифр мантиисы
 * @param n длина массива
 * @return int количество цифр
 */
int count_digits(int8_t *arr, size_t n);

/**
 * @brief Округление
 *
 * @param arr массив цифр мантиисы
 * @param n длина массива
 */
void round_mantiss(int8_t *arr, size_t n);

/**
 * @brief Перевод из строки в структуру действительного числа
 *
 * @param real_num_str строка с действительным числом
 * @param real_num действительное число
 * @return int код возврата
 */
int str_to_real_num(char *real_num_str, real_num *real_num);

/**
 * @brief Перевод из строки в структуру целого числа
 *
 * @param integer_num_str строка с целым числом
 * @param integer_num целое число
 * @return int код возврата
 */
int str_to_int_num(char *integer_num_str, int_num *integer_num);

/**
 * @brief Перевод структуры результата в строку
 *
 * @param res_num структура результата
 * @param res_num_str строка результата
 * @return int код возврата
 */
int res_to_str(calc_num res_num, char *res_num_str);

```

```
/**
 * @brief Произведение чисел
 *
 * @param real_num_str строка с действительным числом
 * @param integer_num_str строка с целым числом
 * @return int код возврата
 */
int calculate(real_num real_num, int_num integer_num, calc_num
*res_num);
```

ОПИСАНИЕ АЛГОРИТМА

1. Программа считывает две строки `real_num_str` и `int_num_str`, одна содержит действительное число, другая — целое.
2. Строки переводятся в структуры `real_num` и `int_num`.
3. Перемножение двух чисел по цифрам с записью в структуру `res_num` за $O(n^2)$.
4. Вычисление знака мантиссы и экспоненты структуры `res_num`.
5. Если в мантиссе больше 30 символов, то происходит округление, а откинутая часть увеличивает порядок.
6. Незначащие нули удаляются из мантиссы, а их количество добавляется к порядку.
7. Добавление к порядку количества цифр в мантиссе.
8. Проверка порядка на переполнение.
9. Перевод данных структуры `res_num` в строку в нормализованном виде.
10. Вывод строки

НАБОР ТЕСТОВ

№	Название теста	1-е число	2-число	Вывод
1	На единицу + разные знаки	-123.456E-3	1	-0.123456E+0
2	На ноль	123	0	0 (нуль) непредставим в нормализованном виде.
3	На 10	123.E123	10	+0.123E+127
4	40 символьные на 30 символьное	123...7899	123...7899	+0.152415787532388367504953 515637E+69
5	Округление 31 символьного	999...999 (31)	1	+0.1E+32
6	Округление на 1 разряд	10000000000000000 0000000000000005	1	+0.10000000000000000000000000000 000001E+31
7	Превышение порядка до нормализации	0.001e-99999	2000	+0.2E-99998
8	Превышение порядка	1e99999	1	Ошибка: Произошло переполнение порядка.
9	Ввод слишком длинного числа	9999999999999999 9999999999999999 999999999	-	Ошибка размера мантиссы
10	Пустая строка		-	Пустая строка.
11	Неправильный ввод	abc	-	Введенные данные не соответствуют формату входных данных.
12	Действительное в целое	1	1.0	Введенные данные не соответствуют формату входных данных.

ОЦЕНКА ЭФФЕКТИВНОСТИ

Оценка сложности алгоритма умножения чисел – $O(m*n)$, так как каждая цифра действительного числа умножается на каждую цифру целого числа. Остальные алгоритмы, используемые в программе, являются линейными.

Обоснование выбора структур данных: Мантиссы чисел представлены как массивы чисел типа `int8_t` длиной 40, 30 и 70. Знаки мантисс представлены как числа типа `int8_t`. Порядок представлен типом `int`, т.к. согласно задаче, порядок не превышает 99999 по модулю, поэтому можно обойтись встроенным типом данных. Тип `int8_t` выбран, потому что в нем будут храниться значения от 0 до 9 и при этом является наименьшим типом чисел, его размер всего 1 байт, что позволяет эффективно использовать память при использовании данных структур.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от разрядности процессора и выбранного типа переменной. Максимальный диапазон целого числа имеет 64-разрядное целое число типа `long long`, который представляет целое число в диапазоне от $-9\,223\,372\,036\,854\,775\,808$ до $+9\,223\,372\,036\,854\,775\,807$.

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

Точность представления вещественных чисел определяется количеством памяти, выделяемой для хранения мантиссы числа. Например, для мантиссы числа типа `double` выделяется 52 бита, с помощью этого мантисса числа может иметь значение до $4\,503\,599\,627\,370\,496$.

3. Какие стандартные операции возможны над числами?

Стандартные арифметические операции в языке СИ: Сложение, вычитание, умножение, деление, взятие остатка, инкрементирование, декрементирование, изменение знака.

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

В таком случае программисту потребуется создать свой тип данных — структуру, содержащую например мантиссу и её знак и порядок и его знак.

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

Можно использовать самостоятельно разработанные функции или воспользоваться готовыми библиотеками.

Вывод

При выполнении данной лабораторной работы по обработке больших чисел, мне довелось рассмотреть множество вариантов реализаций структур данных, способных представлять большие числа. В результате, была выбрана подходящая структура для поставленной задачи и написана программа, совершающая ввод и вывод данных, преобразование строк в структуры и саму операцию умножения над длинными числами, пользуясь представленным в методических материалах алгоритмом перемножения длинных чисел.