



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

Студент **Паламарчук А.Н.**

Группа **ИУ7-33Б**

Предмет **Типы и структуры данных**

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

Студент _____ **Паламарчук А.Н.**

Руководитель _____ **Никульшина Т. А.**

2023 г.

Описание условия задачи

Написать программу для моделирования операции умножения двух длинных чисел.

Техническое задание

Смоделировать операцию умножения целого числа длиной до 40 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 40 значащих цифр, а $K1$ - до 5 цифр.

Входные данные

- Строка, в которой задано целое число длиной до 40 значащих разрядов с необязательным указанием знака
- Строка, в которой задано действительное число в экспоненциальной форме с длиной мантиссы до 30 значащих разрядов с необязательным указанием знака мантиссы

Выходные данные

- Строка, в которой записан результат перемножения входных чисел, с длиной мантиссы до 40 значащих разрядов и порядком до 5 цифр

Возможные аварийные ситуации

- Некорректный ввод исходных данных
- Переполнение порядка

Описание внутренних структур данных

Структура данных называется `bignum_t` и выглядит следующим образом

```
typedef struct {  
    unsigned char is_negative;  
    unsigned char mantissa_field[MANTISSA_LEN];  
    long order;  
} bignum_t;
```

Первое поле используется для хранения знака мантиссы.

Второе поле – массив типа `unsigned char` длиной 40, используемый для хранения мантиссы числа.

Тип `unsigned char` был выбран для того, чтобы данный массив занимал наименьшее количество памяти

Третье поле используется для хранения порядка и её знака. Тип данных `long` был выбран для того, чтобы гарантировать достаточное количество памяти для хранения значения порядка и её знака.

Входные числа и результат вычислений хранятся в переменных этой структуры данных.

Описание заголовочного файла

Константы

```
#define MANTISSA_LEN 40
#define MAX_ORDER 99999
#define MIN_ORDER -99999
```

Ошибки

```
#define ERROR_INPUT_STRUCT 1
#define ERROR_ADD 2
#define ERROR_INPUT_FIRST 3
#define ERROR_INPUT_SECOND 4
#define ERROR_MUL 5
```

Используемые функции

```
void bignum_init(bignum_t *a);
```

Инициализирует 0 переданную структуру.

```
void bignum_print_field(unsigned char *a, size_t alen);
```

Поле мантиссы хранится следующим образом 0000001234. Данная функция распечатывает поле в валидном виде.

```
void bigint_print(bignum_t *a);
```

Распечатывает переданную структуру, хранящую целое.

```
void bigfloat_print(bignum_t *a);
```

Распечатывает переданную структуру, хранящую действительное.

```
void revers_num(unsigned char *a, size_t alen);
```

Разворачивает массив 000004321 в 000001234.

```
int sbigint_scanf(bignum_t *a);
```

Считывает из стандартного потока ввода длинное целое.

```
int sbigfloat_scanf(bignum_t *a);
```

Считывает из стандартного потока ввода длинное действительное.

```
int is_null(unsigned char *a, size_t alen);
```

Проверяет пустоту мантиссы.

```
size_t len_mantiss(unsigned char *str, size_t max_len);
```

Считает длину мантиссы.

```
int bignum_add_field(unsigned char *dst, unsigned char *src, size_t len);
```

Операция сложения двух массивов, нужно для умножения в столбик.

```
int append_zero(unsigned char *a, int alen, size_t count_zero);
```

Дополняет мантиссу указанным кол-вом нулей.

```
int bignum_field_mul_digit(unsigned char *dst, unsigned char *src, size_t maxlen, int digit);
```

Осуществляет умножение массива цифр(мантиссы) на цифру.

```
int bignum_round(unsigned char *dst, int dlen, unsigned char *src, int slen, int len_mantiss);
```

Осуществляет округление массива цифр(мантиссы) до нужной длины

```
int bignum_mul(bignum_t *a, bignum_t *b, bignum_t *res);
```

Осуществляет операцию умножения.

Описание алгоритма

1. Если хотя бы одно из чисел равно нулю в качестве результата возвращается структура типа `bignum_t` с нулевыми значениями мантиссы и экспоненты, а выполнение алгоритма завершается.
2. Создаётся временный массив для хранения промежуточных результатов вычислений, который заполняется нулями. Массив имеет длину равную сумме длин мантисс целого и действительного и тип `unsigned char`, так как при умножении двух чисел, наибольшая длина мантиссы, которая может быть получена равна сумме длин мантисс исходных чисел
3. Выполняется немного изменённое умножение «столбиком» над входными числами.
 - 3.1. Очередная цифра второго числа (цифры берутся по порядку с конца) умножается на все цифры первого числа циклически, а результаты умножений прибавляются к соответствующим ячейкам созданного ранее временного массива.
 - 3.2. Выполняется перемещение переполненных разрядов (если в ячейке временного массива число превышает 9, в ячейку с индексом на 1 меньше прибавляется число, равное результату целочисленного деления значения данной ячейки на 10, а в самой ячейке остаётся лишь остаток от деления хранимого числа на 10). Пройдя весь временный массив от конца к началу, мы избавимся от всех переполнений и сможем снова перейти к пункту 3.1, пока значащие разряды во втором числе не закончатся.
4. Формируется новая мантисса, соответствующая пределам, заданным в техническом задании (то есть 40 значащих разрядов)
 - 4.1. Определяется округление для 40-ой значащей цифры
 - 4.2. Проходим по оставшимся 40 цифрам при переполнении остаток от деления переходит к следующему элементу
5. Вычисляется новый разряд
6. Разряд проверяется на переполнение.

7. Производится запись мантиссы в выходную переменную с помощью вычисленных значений первого и последнего индексов временного массива
8. Производится запись разряда в выходную переменную

Тестирование программы

Аварийные ситуации

Программа осуществляет умножение целого числа на действительное с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

----|----1----|----2----|----3----|----4----|----5

а

Ошибка ввода первого числа, типа 1

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

----|----1----|----2----|----3----|----4----|----5

1234a

Ошибка ввода первого числа, типа 3

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

----|----1----|----2----|----3----|----4----|----5

123

Введите действительное число в формате +-m.n E +-K,

где суммарная длина мантиссы (m+n) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

----|----1----|----2----|----3----|----4----|----5

1234.1234г

Ошибка ввода второго числа, типа 5

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

Программа осуществляет умножение целого числа на действительное с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков. Введите целое число в формате $\pm m$, где длина мантиссы m – до 40 значащих цифр:

Введите действительное число в формате $\pm m.n \text{ E } \pm K$,
где суммарная длина мантиссы ($m+n$) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

Программа осуществляет умножение целого числа на действительное с длиной мантииссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков. Введите целое число в формате +m, где длина мантииссы m – до 40 значащих цифр:

Введите действительное число в формате $+m.n \text{ E } +K$,
где суммарная длина мантииссы ($m+n$) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

Программа осуществляет умножение целого числа на действительное с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков. Введите целое число в формате +m, где длина мантиссы m – до 40 значащих цифр:

Введите действительное число в формате $\pm m.n \text{ E } \pm K$,
где суммарная длина мантиссы ($m+n$) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

```

----|----1----|----2----|----3----|----4----|----5
12345 E -100000

```

-----1-----2-----3-----4-----5

0.0 E 0

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

-----1-----2-----3-----4-----5

1234567

Введите действительное число в формате +-m.n E +-K,

где суммарная длина мантиссы ($m+n$) – до 30 значащих цифр, а величина порядка k – до 5 цифр:

-----1-----2-----3-----4-----5

0.0000 E 0

-----|-----1-----|-----2-----|-----3-----|-----4-----|-----5-----

$$0.0 \in 0$$

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m,

где длина мантиссы m – до 40 значащих цифр:

-----1-----2-----3-----4-----5

9000000000000000000000000000000000009

Введите действительное число в формате +-m.n E +-K,

где суммарная длина мантиссы ($m+n$) – до 30 значащих цифр, а величина порядка k – до 5 цифр:

-----1-----2-----3-----4-----5

1000000000000000000000000000000001

-----1-----2-----3-----4-----5

0.9000000000000000000000000000000018 E 59

Программа осуществляет умножение целого числа на действительное

с длиной мантиссы целого до 40 знаков и действительного до 30 с порядком до 5 знаков.

Введите целое число в формате +-m.

где длина мантиссы m – до 40 значащих цифр:

-----1-----2-----3-----4-----5

999

Введите действительное число в формате +-m.n E +-K.

где суммарная длина мантиссы ($m+n$) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

Введите действительное число в формате +-m.n E +-K,

-1000000000000000

Введите действительное число в формате +-m.n E +-K,

где суммарная длина мантиссы (m+n) – до 30 значащих цифр, а величина порядка K – до 5 цифр:

----|----1----|----2----|----3----|----4----|----5

1 E 0

----|----1----|----2----|----3----|----4----|----5

-0.1 E 14

Выводы по проделанной работе

Для работы с длинными числами, выходящими за разрядную сетку ПК нужно использовать структуры данных, в которых будут храниться части этих чисел. Арифметические операции необходимо адаптировать для работы с нашим типом данных. Умножение длинных чисел происходит поразрядно.

Контрольные вопросы

Каков возможный диапазон чисел, представляемых в ПК?

Точность представления чисел зависит от объёма памяти, выделенного переменной и наличию знака у неё. Для ЧПТ от 3.6E-4951 до 1.1E+4932, длина мантиссы, согласно стандарту IEEE754 и переменной типа double языка Си – от 15 до 16 значащих разрядов, длину мантиссы у целых чисел проще всего посчитать, выведя максимальное значение данного числа, но можно взять в качестве примера тип long long языка Си – 19 разрядов (или 20, если тип беззнаковый)

Какова возможная точность представления чисел, чем она определяется?

Диапазон зависит от разрядности процессора и операционной системы.

Для целых чисел ограничением является $2^n - 1$, где n – количество бит, выделенных для данной переменной, в случае целых со знаком – имеем ограничение от $-2^{(n-1)}$ до $2^{(n-1)} - 1$ (-1 для нуля)

Для действительных чисел имеем абсолютное ограничение (по модулю) от

3.6E-4951 до 1.1E+4932, длина мантиссы, согласно стандарту IEEE754 и переменной типа double языка Си – от 15 до 16 значащих разрядов (в 10-й СС)

Какие стандартные операции возможны над числами?

Для каждого числа должны быть определены арифметические операции (сложение, вычитание, деление, умножение) и операции сравнения. Также существуют логические и побитовые операции.

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

Программист может выбрать свой тип, основным элементом которой будет являться массив цифр.

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

Такие операции можно осуществить поразрядно.