|  |  |
| --- | --- |
|  | **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 К ЛАБОРАТОРНОЙ РАБОТЕ №1 ПО «ТИПАМ И СТРУКТУРАМ ДАННЫХ»**

**Тема: «Длинная арифметика»**

Группа: ИУ7-32Б

|  |  |  |
| --- | --- | --- |
| Студент: |  | Сидоров Максим Михайлович |

Преподаватель: Никульшина Татьяна Александровна

**Постановка задачи:** смоделировать операцию умножения действительного числа, мантисса которого имеет не более 30 знаков на целое число, длина которого не более 30 знаков и вывод числового результата на экран в нормализованной форме.

**Описание ТЗ:**

1. Входные данные:

Действительное число должно быть представлено в форме ‘±m.n Е ±K’, где суммарная длина мантиссы (m+n) - до 30 значащих цифр, а величина порядка K - до 5 цифр.

Целое число должно быть представлено в форме ‘±m’, где m до 30 значащих цифр.

1. Выходные данные: результат умножения в форме ‘±0.m Е ±K’, где m до 30 значащих цифр, а величина порядка K - до 5 цифр.
2. Программа выполняет умножение действительного числа в указанной форме на целое число в указанной форме и выдает результат в форме ‘±0.m Е ±K’, где m - до 30 значащих цифр, а K - до 5 цифр.
3. Чтобы запустить программу, следует из папки программы запустить скрипт “./build\_release.sh”, который скомпилирует программу и создаст исполняемый файл, а затем из этой же папки запустить сам исполняемый файл, указав команду “./app.exe”.
4. При пользовательском вводе, отличном шаблону, описанному выше, будет выведено на экран сообщение «Некорректные данные». Также при выходе порядка ответа за границу промежутка (-99999;99999) на экран будет выведено сообщение «Произошло переполнение».

**Описание СД:**

1. Для хранения знака чисел было выбрано следующее представление:

enum

{

plus = 1,

minus = -1

} sign;

*Листинг 1. Представление знака чисел.*

1. Для действительного числа выбрана следующая структура:

struct {

sign sign;

unsigned short mantissa[30];

unsigned short mantissa\_len;

int power;

} my\_double;

*Листинг 2. Представление действительного числа.*

В данной структуре хранятся: знак числа; сами цифры действительного числа и ведущие нули, если количество знаков мантиссы <30; количество значащих цифр мантиссы; порядок числа.

Например, число +72364.2342 Е -324 будет храниться как:

sign = plus, mantissa[30] = 000000000000000000000723642342,

mantissa\_len = 9, power = -324.

1. Для хранения целого числа выбрано аналогичное действительному представление, за исключением порядка:

struct

{

sign sign;

unsigned short mantissa[30];

unsigned short mantissa\_len;

} my\_int;

*Листинг 3. Представление целого числа.*

**Описание алгоритма:**

1. Сохранение чисел в указанные СД.
2. Реализация подсчетов производится способом, похожим на способ вычисления в столбик. Вычисление мантиссы: начиная с последней цифры одного из чисел последовательно умножать на каждую цифру другого числа и записывать в массив сумм, похожий на вычисления в столбик, так делать пока не переберем все цифры первого числа. Полученный массив и будет результатом.

Например:

00000000000000000000000000123 \* 00000000000000000000000000543

Составим алгоритм вычисления:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | … | 0 | 0 | 6 | 4+1 | 1+3+8+5 | 6+2 | 9 |
| 0 | 0 | … | 0 | 0 | 6 | 5+1 | 17-10 | 8 | 9 |
| 0 | 0 | … | 0 | 0 | 6 | 6 | 7 | 8 | 9 |

*Таблица 1. Алгоритм подсчета произведения.*

000000000000000000000000066789. Это и есть результат.

1. Вычисление знака: если у исходных чисел одинаковый знак, то у результата знак ‘+’ иначе ‘-‘.
2. Длина новой мантиссы = (30 – количество ведущих нулей в массиве)
3. Новый порядок = (порядок действительного числа + длина мантиссы ответа – длина мантиссы действительного числа)

**Описание функций:**

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Int check\_double\_mantissa(my\_double \*a, int \*add\_power, const char \*s) | Запись мантиссы действительного числа |
| Int check\_int\_mantissa(my\_int \*a, const char \*s) | Запись мантиссы целого числа |
| Bool check\_double\_power(my\_double \*a, const char \*s, int add\_power) | Запись порядка действительного числа |
| Bool check\_double\_sign(my\_double \*a, const char \*s) | Запись знака действительного числа |
| Bool check\_int\_sign(my\_int \*a, const char \*s) | Запись знака целого числа |
| Int get\_double(my\_double \*a, const char \*s) | Запись действительного числа из строки в структуру |
| Int get\_int(my\_int \*a, const char \*s) | Запись целого числа из строки в структуру |
| Void count(const my\_double \*a, const my\_int \*b, my\_double \*answer) | Подсчет произведения |
| Bool print\_new\_double(const my\_double \*a) | Вывод ответа |

**Тесты:**

Для первого теста будет представлен полный вывод программы, для остальных – частичный.

Программа предназначена для вычисления произведения целого числа и действительное.

Укажите действительное число в формате "(+-)m.n E (+-)K"; знаки '+' или '-' ставить обязательно, букву Е латинскую, которая разграничивает порядок, ставить обязательно, пробелы ставить в соответствии с образцом. Также m+n <= 30 и К - целое число из промежутка (-99999;99999).

| | | |

Введите действительное число: +122 E -3

Укажите целое число в формате "(+-)m"; знак '+' или '-' ставить обязательно. Также m <= 30.

| | | |

Введите целое число: -2

Произведение: -0.244 E +0

*Тест 1. Корректные данные.*

Введите вещественное число: +324g E -23

Некорректные данные

*Тест 2. Некорректные данные для действительного числа.*

Введите действительное число: +234.234 E -23

Введите целое число: +324h24

Некорректные данные

*Тест 3. Некорректные данные для целого числа.*

Введите действительное число: +0.0 E -23

Введите целое число: +2

Произведение: +0.0 E +0

*Тест 4. Нулевое вещественное число.*

Введите действительное число: +345.543 E -3

Введите целое число: +0

Произведение: +0.0 E +0

*Тест 5. Целое число нулевое.*

*Тест 8. Порядок ответа <-100000.*

Введите действительное число: +99999999999999999999999999999 E +99999

Введите целое число: +9999999999999999999999999999

Произошло переполнение

*Тест 6. Оба числа нулевые.*

Введите действительное число: +0.0 E -3

Введите целое число: +0

Произведение: +0.0 E +0

*Тест 7. Порядок ответа >100000.*

Введите действительное число: +0.000999999999999999999999999 E -99999

Введите целое число: +1

Произошло переполнение

Введите действительное число: +23475628734.248762782346827 E -2343

Введите целое число: -273647823648762387462378362487

Произведение: -0.642405471191352442064427080647 E -2303

*Тест 9. Длинные числа.*

Введите действительное число: +999999999999999999999999999999 E -23

Введите целое число: +2

Произведение: +0.2 E +8

*Тест 10. Циклическое округление.*

Введите действительное число: +888888888888888888888888888888 E -2

Введите целое число: +2

Произведение: +0.177777777777777777777777777778 E +29

*Тест 11. Округление.*

**Вывод:** умножение чисел, размер которых превосходит разрядную сетку компьютера программист реализует самостоятельно. Это можно сделать на основе массива, состоящего из цифр этого числа, с использованием дополнительных файлов и другими способами. Я использовал массивы, состоящие из цифр. При разработке программы самым сложным было придумать алгоритм подсчета произведения, с чем я успешно справился. Эта лабораторная работа помогла мне понять, как работает длинная арифметика и как хранятся числа в памяти компьютера.

**Ответы на вопросы:**

1. **Каков возможный диапазон чисел, представляемых в ПК?**

Диапазон чисел зависит от разрядности процессора. На современных ПК разрядность составляет 64 бит, следовательно диапазон числе укладывается в промежуток [-2^63;2^63 - 1].

1. **Какова возможная точность представления чисел, чем она определяется?**

Точность представления числа определяется длиной его мантиссы, то есть от выделенного количества памяти под мантиссу, ее типа, знака и порядка. У чисел типа double на точность отводится 53 бита, что соответствует примерно 16 десятичным знакам.

1. **Какие стандартные операции возможны над числами?**

Сложение, вычитание, умножение, деление и остаток от деления.

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

Для таких чисел программист может выбрать массив и хранить в каждой ячейке цифру исходного числа.

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

Можно разбить число на цифры и записать в массив, а затем выполнять операции с массивом цифр.