



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»
по курсу «Типы и структуры данных»

Студент Чепиго Дарья Станиславовна
Группа ИУ7-44Б

Студент:

_____ Чепиго Д.С..
подпись, дата Фамилия, И. О.

Преподаватель:

_____ Барышникова М.Ю..
подпись, дата Фамилия, И. О.

Оценка _____

Москва – 2021 г.

Условие задачи

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

Техническое задание

Входные данные

Одно действительное десятичное число, записанное в формате $[\pm]mmm.nnn[e][\pm]ppppp$ (в квадратных скобках даны варианты символа на одной и той же позиции), где m , n , p – цифры, причем суммарное количество цифр до и после точки не превышает 30, число цифр после знака порядка не превышает 5, а сам порядок по модулю меньше, чем 99999, буква «e» – латинские, знак «+» или «-» у мантиссы и у порядка должен обязательно присутствовать. Ведущие нули в целом числе включаются в подсчет длины числа. Одно целое десятичное число, состоящее не более чем из 30 цифр, перед которыми обязательно должен быть знак «+» или «-».

Выходные данные

Одно действительное число, записанное в формате $[\pm]0.mmmm[E/e][\pm]ppppp$ (в квадратных скобках даны варианты символа на одной и той же позиции), где m , p – цифры, причем количество цифр в мантиссе не превышает 30, знак порядка может присутствовать, число цифр после знака порядка не превышает 5.

Способ передачи входных данных

Входные данные передаются программе путем ввода в стандартный поток ввода на соответствующий запрос программы.

Способ получения выходных данных

Выходные данные выводятся в стандартный поток вывода.

Задача программы

Программа, получив на вход два числа, осуществляет их умножение и выводит результат в указанном формате. При невозможности получения результата программа выводит сообщение об ошибке.

Способ обращения к программе

Обращение к программе осуществляется путем запуска исполняемого файла программы `app.exe`.

Возможные ошибочные ситуации

1. Отсутствие одного или двух чисел в стандартном потоке ввода;
2. Некорректный ввод одного или двух чисел;
3. Переполнение порядка результата.

Описание используемых структур данных

Строка, содержащая число

Содержит символы, введенные пользователем. Для целого числа длина такой строки равна 32 (знак числа, 30 цифр, символ перехода на новую строку «\n» и символ конца строки «\0»), для вещественного – 41 (знак мантииссы, 30 цифр в мантииссе, точка в мантииссе, знак порядка, 5 цифр в порядке, символ перехода на новую строку «\n» и символ конца строки «\0»).

```
char real_num[MAX_STR]; - строка для действительного числа (MAX_STR = 41)
char int_num[MAX_INT]; - строка для целого числа (MAX_INT = 33)
```

Массив цифр

Для хранения результата произведения целого числа и мантииссы действительного числа используется массив цифр. Массив имеет длину 60. Это сделано для того, чтобы обеспечить корректность вычислений при переходе через десяток, когда длина мантииссы максимально допустимая. Каждая цифра имеет тип int, в связи с используемым алгоритмом. Цифры записываются в массив справа налево, то есть последняя цифра числа имеет индекс 0, предпоследняя 1 и т. д. Это сделано для удобства вычислений.

```
int result[MAX_MULTII] = {0}; - массив для произведения двух чисел
(MAX_MULTII = 60)
```

Структура «длинное вещественное»

Содержит строку, содержащую все число (со знаками, точками и символом экспоненты), знак мантииссы типа char, массив цифр мантииссы типа char, знак порядка типа char, значение порядка типа int и положение точки типа size_t. Данные хранятся в одной структуре для обеспечения удобной совместной обработки всех данных.

```
struct real_number
{
    char real_num[MAX_STR]; - строка для действительного числа (MAX_STR = 41)
    char sign_mantissa; - символ знака мантииссы
    char mantissa[MAX_MANTISSA]; - строка для цифр мантииссы (MAX_MANTISSA = 32)
    char sign_order; - символ знака порядка
    int order; - массив, содержащий порядок
    size_t point_place; - позиция точки в мантииссе
};
```

Структура «длинное целое»

Данная структура содержит знак целого числа типа char, массив его цифр типа char. Данные хранятся в одной структуре для обеспечения целостной обработки данных.

```
struct int_number
{
    char sign_int; - символ знака числа
    char int_num[MAX_INT]; - массив, содержащий целое число (MAX_INT = 33)
};
```

Описание алгоритма

1. Ввод данных

При запуске программы на экран выводятся правила, которые надо соблюдать при вводе чисел. Если хотя бы одно из них не выполнено – программа выдаст ошибку. Ниже – пример программы с правилами и удобным примером для ввода.

```
Rules for input:
When you enter the real number, please:
0.Enter elements of number without space
1.Enter the sign of mantissa(+ or -)
2.After the first point enter the mantissa(max 30 numbers,necessarily 1 point,
you must enter the number before point)
3.After the second point enter the symbol of exponenta - 'e'
4.After the third point enter the sign of order(+ or -)
5.After the fourth point enter the order(max 5 numbers)
6.Click enter

When you enter the integer number, please:
0.Enter elements of number without space
1.Enter the number(max 30 numbers) with sign of number(+ or -)
2.Click enter

+.....e+.....
Enter the real number:+1111122222333334444455555.66666e+99999

+.....
Enter the integer number:+111112222233333444445555566666
```

Считываем действительное и целое число в соответствующие строку из стандартного потока ввода. Если они считались успешно, то далее из строк записываем данные в соответствующие элементы структуры.

2. Запись во внутренние структуры

Обработывая строку действительного числа по очереди проверяется все её содержимое, а именно наличие знаков, символа экспоненты, корректность мантииссы и порядка, находится позиция точки. После этого высчитывается порядок чисел и осуществляется нормализация. Далее удаляется точка в массиве чисел мантииссы для алгоритма умножения.

Целое число аналогично проверяется на наличие знака и корректные цифры. Запоминается знак, цифры записываются в массив символов. Если в ходе обработки входных данных произошла ошибка, выводим сообщение и завершаем работу программы.

3. Вычисления

Умножение двух чисел

Для алгоритма умножения двух чисел, размер которых максимум 30 цифр создаётся массив на 60 символов. Это реализуется с помощью двух циклов for с переменной i для первого числа и переменной j для второго числа соответственно. По очереди все цифры первого числа умножаются на все числа второго числа, их произведение записывается в ячейку массива под номером (60 – i - j). При этом после завершения этих циклов происходит проход по массиву с конца для переноса чисел, которые больше 10 в следующий разряд соответственно.

Вычисление порядка

После того, как мы знаем количество значащих цифр в результате нашего умножения мы можем вычислить порядок по формуле $\text{=количество значащих цифр в результате} - \text{изначальный порядок действительного числа} - \text{количество цифр в действительном числе после точки}$. Если при округлении первой цифрой числа была “9” и это округлилось, то порядок меняется на 1.

Корректировка мантиссы

Число знаков в мантиссе может превысить максимально допустимое. Найдем разницу между числом знаков результата и максимально допустимым. Если оно не положительно, корректировка не требуется. В противном случае сдвинем разряды результата так, что первая цифра займет 30-е справа место, предварительно запомнив цифру, идущую после первой справа сохраняемой цифры. Если запомненная цифра от 5 до 9, то прибавим 1 в последнем разряде мантиссы.

Определение знака

Если два исходных числа имеют одинаковый знак, знак результата принимается равным «+», иначе – «-».

4. Вывод результата или сообщения

Если в ходе обработки входных данных произошла ошибка, выводим сообщение и завершаем работу программы. Если ошибки нет, выводим результат в указанном выше формате.

Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Если под хранение целого положительного числа выделено 16 разрядов, то его максимальное значение не может превышать $2^{16} - 1 = 65\,535$, если выделено 32 разряда, то максимальное значение составит $2^{32} - 1 = 4\,294\,967\,295$. Для 64 разрядов максимально возможное значение числа равно $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$.

Вещественные числа представляются в виде мантиссы и порядка. Мантисса – правильная дробь, первая цифра которой отлична от нуля, т.е. находится в интервале $[0.1, 1)$. Порядком – степень основания системы счисления, в которой сохранено число. Длина мантиссы определяет точность представления числа, а длина порядка ограничивает диапазон допустимых значений. Максимально под представление мантиссы отводится 52 разряда в двоичной системе счисления, а под представление порядка – 11 разрядов. В этом случае возможные значения чисел находятся в диапазоне от $3.6 \cdot 10^{-4951}$ до $1.1 \cdot 10^{+4932}$.

2. Какова возможная точность представления чисел?

Целые числа хранятся в памяти ПК точно. Вещественные числа могут быть: одинарной точности, тогда точность составляет 7-8 значащих цифр после запятой; двойной точности с 15-16 значащими цифрами; четверной точности – около 34 значащих цифр.

3. Какие стандартные операции возможны над числами?

Набор операций над числом определяется его типом. Целые и вещественные числа поддерживают арифметические операции (сложение, вычитание, умножение, деление) и операции сравнения (кроме сравнения на равенство для вещественных чисел, их надо сравнивать, учитывая погрешность – некоторое маленькое число).

Целые числа в языках подобных Си также поддерживают битовые операции (битовые И, ИЛИ, ИЛИ-НЕ, НЕ) и операцию взятия остатка от деления, отдельная операция для целочисленного деления. В некоторых языках программирования, реализована операция возведения в степень.

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

Программист может сохранить цифры числа в массиве или разбить число на части, каждую часть сохранить как отдельное число, обработать эти части и затем склеить их для выдачи результата пользователю.

5. Как можно осуществить операции над числами, выходящими за рамки машинного представления? Можно сохранить числа в самостоятельно разработанном представлении, как в 4 вопросе, и обработать сохраненные таким образом числа с использованием операций над используемыми типами данных. Для четырех арифметических операций – сложения, вычитания, умножения, деления – существует алгоритм вычисления результата «столбиком», который несложно реализовать самим. Для операций сравнения можно использовать поразрядное сравнение.

Тестирование

Тут должно быть тестирование, но оно в 1 лабе было огромное, так как я скринила ВСЕ func тесты, которые выглядели как в си...

Собственно так делать не надо и копировать сюда я это тоже не стала :з

Выводы

По итогам лабораторной работы заключаем следующее:

1. Целые и вещественные числа имеют различное хранение в памяти компьютера: целые – точно, а вещественные – приближенно (точность зависит от используемого типа данных).
2. Невозможно сохранить сколь угодно большое (маленькое) по модулю число со сколь угодно большой точностью.
3. В случае, если не удастся использовать стандартные типы данных, то программист может самостоятельно решать этот вопрос разными способами.