|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04** Программная инженерия

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 1 |

**Название:**

Обработка больших чисел

**Дисциплина:** Типы и структуры данных

|  |  |  |  |
| --- | --- | --- | --- |
| Студент | ИУ7-31Б |  | Н. Ю. Баринов |
|  | (Группа) |  | (И.О. Фамилия) |
| Преподаватель | Барышникова Марина Юрьевна |  |  |

Москва, 2021

**Описание условия задачи**

Смоделировать операцию умножения целого числа длиной до 30 десятичных цифр на действительное число в форме ±m.n Е ±K, где суммарная длина мантиссы (m+n) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Описание ТЗ**

1. **Описание исходных данных**

на вход поступает строка в формате +/-XX.XX, затем порядок (Е) +/-AAA и строка в формате +/-YY

X - цифры, содержащиеся в мантиссе

A - цифры, содержащиеся в порядке

E - обозначения порядка

Y – цифры целого числа

Ограничения:

- Количество цифр в целом числе не должно превышать 30.

- Количество цифр в мантиссе не должно превышать 30.

- Количество цифр в порядке не должно превышать 5.

- Для корректной работы строки не должны содержать некорректные символы, которые нельзя преобразовать в вещественное число. Далее речь пойдёт о допустимых символах.

- Первым символом допускается указание знака вещественного числа. Если знак отсутствует, то по умолчанию число считается положительным. Это замечание касается и знака порядка.

Несколько знаков + или - в мантиссе, либо в порядке считаются некорректным вводом

**+123 - Допустимый ввод**

**-123 - Допустимый ввод**

**123 - Допустимый ввод**

**+123 E +123 - Допустимый ввод**

**+-123 - Недопустимый ввод**

**-12-3 - Недопустимый ввод**

**+123+ E +123 - Недопустимый ввод**

- В мантиссе можно использовать «.» и «,»

Разрешено, не указывать их.

Дублирование этих символов рассматривается как некорректный ввод.

**123.2 - Допустимый ввод, число будет обработано как (0.1232E3)**

**123,2 E 12 - Допустимый ввод, число будет обработано как (0.1232E15)**

**1232E12 - Допустимый ввод, число будет обработано как (0.1232E16)**

**123 - Допустимый ввод, число будет обработано как (0.123E3)**

**12 E 12E12 - Недопустимый ввод**

**12.12.12 - Недопустимый ввод**

- Допустимо использование незначащих нулей в записи мантиссы и экспоненты, они будут обработаны корректно, но на этапе ввода строки могут вызвать ошибку переполнения ввода.

Знак следует указывать до незначащих нулей.

**000000123 - Допустимый ввод, число будет обработано как (0.123E3)**

**000000000000000000000000000000000012 - Недопустимый ввод, общая длина строки**

**превышает максимально допустимую (STRING\_SIZE)**

**+0012 - Допустимый ввод, число будет обработано как (0.12E2)**

**00+12 - Недопустимый ввод.**

- Максимальным значением порядка является 99999, минимальным -99999

- Использование нечисловых символов, кроме «.», «,», «+», «-». Рассматривается как некорректный ввод

**123kjjkasd - Недопустимый ввод**

**asd..asdasdE - Недопустимый ввод**

1. **Описание результатов программы**

В результате выполнения программы будет выведено вещественное число в формате

+/-XX.XX\*\*E\*\*+/-AAA

X - цифры, содержащиеся в мантиссе

A - цифры, содержащиеся в порядке

E - обозначения порядка

- Вещественное число имеет максимальный размер мантиссы 30 и максимальное значение порядка 99999 (минимальное -99999). Если результат деления будет иметь более 30 цифр в мантиссе, это значение округляется (Если 31 цифра >= 5, то значение 30 цифры увеличивается на 1, иначе 31 цифра игнорируется).

**Описание задачи, реализуемой в программе**

Программа реализует умножение вещественного числа на целое число.

Алгоритм решения схож с алгоритмом умножения "в столбик"

**Способ обращения к программе**

Обращение к программе происходит через консоль, путём запуска exe файла.

Исполняемый файл имеет расширение .exe, что предусматривает запуск программе в ОС Windows

**Описание возможных аварийных ситуаций и ошибок пользователя**

Часть ошибочных ситуаций описана в заголовке

Описание исходных данных

Возможный вывод программы в случае ошибки:

- float ошибка ввода - произошёл некорректный ввод вещественного числа

- int ошибка ввода - произошёл некорректный ввод целого числа

- Переполнение порядка – произошел выход порядка числа за границы допустимого

**Описание внутренних структур данных**

1. Программа содержит в себе структуру, представляющую вещественное число

typedef struct

{

    my\_sign sign; *// знак*

    short int mantissa[MAX\_MANTISSA]; *// мантисса*

    short int mantissa\_len; *// длина мантиссы*

    int exponent; *// порядок*

} my\_float;

**sign** - является перечисляемой переменной, указывающей на знак числа.

Если значение **sign** == plus, то число является положительным.

Если значение **sign** == minus, то число является отрицательным.

**mantissa** - является массивом целочисленных переменных длинной 65.

Используется для хранения элементов мантиссы

**mantissa\_len** - является целочисленной переменной, указывающей на текущий размер мантиссы

**exponent** - является целочисленной переменной. Используется для хранения порядка.

1. Программа содержит в себе структуру, представляющую целое число

typedef struct

{

    my\_sign sign; *// знак*

    short int number[MAX\_NUMBER]; *// числа*

    short int number\_len; *// длина числа*

} my\_int;

**sign** - является перечисляемой переменной, указывающей на знак числа.

Если значение **sign** == plus, то число является положительным.

Если значение **sign** == minus, то число является отрицательным.

**number** - является массивом целочисленных переменных длинной 65.

Используется для хранения элементов числа

**number\_len** - является целочисленной переменной, указывающей на текущий размер числа

**Тесты**

**Положительные тесты**

1. Один множитель равен 0
2. В результате максимальное значение порядка
3. В результате минимальное значение порядка
4. Первое число имеет и символ "."
5. Первое число имеет и символ ","
6. Умножение на 1
7. Умножение положительного числа на положительное
8. Умножение положительного числа на отрицательное
9. Умножение отрицательного числа на положительное
10. Умножение отрицательного числа на отрицательное
11. Отсутствует знак у первого числа
12. Отсутствует знак у второго числа
13. Присутствует знак у всех чисел
14. Отсутствует знак у всех чисел
15. Результат округляется в меньшую сторону
16. Результат округляется в большую сторону

**Негативные тесты**

1. Первое число содержит недопустимые символы
2. Второе число содержит недопустимые символы
3. Первое число превышает максимальный допустимый размер
4. Второе число превышает максимальный допустимый размер
5. Первое число содержит лишние символы "\*\*.\*\*"
6. Второе число содержит лишние символы "\*\*.\*\*"
7. Первое число содержит лишние символы "+"
8. Второе число содержит лишние символы "-"
9. Пустая строка вместо первого числа
10. Пустая строка вместо второго числа
11. Значение множителей корректны, но значение порядка результата превышает максимальное допустимое значение.
12. Значение числителя и знаменателя корректны, но значение порядка результата меньше, чем минимальное допустимое значение порядка.

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 0 E 0, 1 | +0.0 E 0 |
| 1 E 99998, 1 | +0.1 E 99999 |
| 0.01 E -99998, 1 | +0.1 E -99999 |
| 1.2 E 0, 1 | +0.12 E 1 |
| 1,1 E 0, 1 | +0.11 E 1 |
| 123 E 0, 1 | +0.123 E 1 |
| 3 E 0, 2 | +0.6 E 1 |
| 3 E 0, -2 | -0.6 E 1 |
| -3 E 0, 2 | -0.6 E 1 |
| -3 E 0, -2 | +0.6 E 1 |
| 1.2 E 0, +1 | +0.12 E 1 |
| +1.2 E 0, 1 | +0.12 E 1 |
| +1.2 E 0, +1 | +0.12 E 1 |
| 1.2 E 0, 1 | +0.12 E 1 |
| 999999999999999999999999999999 E 0, 9 | +0.89999999999999999999999999999 E 1 |
| 999999999999999999999999999999 E 0, 2 | +0.2 E 31 |
| Qwe E 0, 1 | Float ошибка ввода |
| 1 Е 0, qd | Int ошибка ввода |
| 99999999999999999999999999999999 Е 0, 1 | Float ошибка ввода |
| 1 E 0, 9999999999999999999999999999999 | Int ошибка ввода |
| 9..9 E 0, 1 | Float ошибка ввода |
| 9 E 0, 1.1 | Int ошибка ввода |
| +9+1 E 0, 1 | Float ошибка ввода |
| 1 E 0, --1 | Int ошибка ввода |
| E 0, 1 | Float ошибка ввода |
| 1 E 1, | Int ошибка ввода |
| 1 E 99999, 99999999999999999999999999 | Переполнение порядка |
| 0.000000000000001 E -99999, 1 | Переполнение порядка |

**Вывод**

В случае работы с большими числами, превышающими размер регистра, программисту нужно самому выбрать и написать тип данных и методы обработки данных.

**Ответы на контрольные вопросы**

1.Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от длины машинного слова. Возможный диапазон целого знакового числа от -2^32 до 2^32-1. Действительные числа от 3.6\*10^4951 до 1.1\*10^4932. Для 64 разрядной системы.

2.Какова возможная точность представления чисел?

Длина мантиссы определяет точность представления числа. Максимально под представление мантиссы отводится 52 разряда.

3.Какие стандартные операции возможны над числами?

Сложение, вычитание, умножение, деление.

4.Какой тип данных может выбрать программист, если обрабатываемые числа

превышают возможный диапазон представления чисел в ПК? Можно использовать массив чисел, для хранения каждой цифры в числе, но для удобства, можно использовать структуру, в которой хранится знак числа, мантисса, порядок и т.д.

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

представления? Операции над большими числами можно выполнять путём последовательного выполнения операций над всеми цифрами, составляющими число, начиная с конца массива цифр.