|  |  |
| --- | --- |
| 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 Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 «Обработка разреженных матриц»**

Студент: Мурзабеков Руслан Муратович

Группа: ИУ7 – 35Б

*2022 г.*

**Задание**

Смоделировать операцию деления действительного числа в форме

±m.n Е ±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 ±0.m1 Е ±mK1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Входные данные:**

В первой строке вводится 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. Во второй строке также вводится 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.

**Требования к входным данным:**

1) Вещественные числа можно вводить в обычном и в экспоненциальном виде. В обычном виде число вводится в формате ±m.n (точка необязательна) (m + n <= 30), а в экспоненциальном ±m.n E±K (m + n <= 30). Все именованные части чисел (m, n, K, N) должны представлять собой корректные натуральные числа или 0 (+0 = -0);

3) ‘E’ или ‘e’ вводятся латинскими буквами

4) При вводе чисел во входной строке отсутствуют пробелы.

**Выходные данные:**

В результате работы программы должно быть выведено действительное число в формате ±m1.n1 E ±K1, где len(m1 + n1) <= 30, len(K1) <= 5.   
 В случае некорректного ввода или невозможности получить ответ в указанном формате будет выведено соответствующее сообщение.

Программа выполняет деление действительного числа на действительное при входных данных, соответствующих формату.

Программа может быть вызвана через консоль или запущена через любую среду разработки С.

**Аварийные ситуации**

В случае аварийной ситуации выводится сообщение о той или иной ошибке.   
 Кроме набора ошибок в результате некорректного ввода (не соответствующего формату) существует так же ошибка:

- Деление на ноль  
 - Невозможность получить ответ (число не может быть записано в указанном формате из-за переполнения порядка).

**Структуры данных**

Для реализации данной задачи был создана структура, состоящая из логической переменной (знак мантиссы), массива целых чисел (цифры мантиссы) и целого числа (порядок)  
 Выбор данной структуры данных обусловлен форматами входных данных — программа хранит в виде строки те данные, которые не может хранить как число.

Typedef struct  
{  
char sign;  
char mantissa[MAX\_MANTISSA\_LEN + 1 + 1];  
int order;  
} exponential;

Поля структуры:

**sign** - знак числа.

**mantissa** - мантисса числа: 30 элементов на саму мантиссу, один для случая, когда длина целого 30(вещественное должно быть больше, итого 31 элемент), и последний элемент для нуль-терминированного символа.

**order** - порядок числа.

**Описание алгоритма**

1. Ввод действительных чисел. В результате из двух введённых строк получается экземпляр указанной структуры данных.   
 При этом происходит проверка корректности входных данных и форматирование действительных чисел — порядок числа изменяется так, чтобы его мантисса стала целой.

2. Форматирование чисел. В результате из двух введённых строк получается экземпляр указанной структуры данных.   
 При этом происходит проверка корректности входных данных и форматирование действительных чисел — порядок числа изменяется так, чтобы его мантисса стала целой.

Также, числа форматируются так, чтобы мантисса вещественного была больше мантиссы целого.

3. Непосредственно операция деления. Действительные числа хранятся в виде двух частей – мантиссы и порядка. Находится множитель от 0 до 9, он записывается в мантиссу результата, от делителя же вычитается произведение делимого и множителя. Деление выполняется до тех пор, пока делимое не равно нулю, или пока мантисса результата не будет содержать 31 цифру. Из порядка вещественного вычитается порядок целого, результат записывается в порядок частного.

4. Форматирование. Округление мантиссы и изменение порядка полученного числа в соответствии с форматом, требуемом в задании.   
 В случае невозможности вывода числа в указанном формате, печатается сообщение о невозможности произвести вычисления.

**Тесты**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | **Проверка на** | **Вещественное число** | | **Целое число** | **Вывод** |
| 1 | Недопустимый символ в первом числе | 32a | |  | Error: incorrect symbol |
| 2 | Два и более знака в мантиссе  в первом числе | ++2 | |  | Error: incorrect sign |
| 3 | Два и более символа ‘e’  в первом числе | 12e12e | |  | Error: too much 'e' |
| 4 | Точка в порядке  в первом числе | 12e0.5 | |  | Error: order must be digit |
| 5 | Знак в конце мантиссы  в первом числе | 1.3-e | |  | Error: incorrect sign |
| 6 | Два и более знака в порядке в первом числе | 12e++3 | |  | Error: incorrect sign |
| 7 | Длина мантиссы более 30  в первом числе | 1111111111111111111111111111111 | |  | Error: mantissa cant be >30 symbols |
| 8 | Длина порядка более 5  в первом числе | 12e100000 | |  | Error: order must be [-99999; 99999] |
| 9 | Некорректный символ во втором числе | 13.45 | | 1q | Error: incorrect symbol |
| 10 | Длина порядка более 5 во втором числе | 345e12 | | 12.3e100000 | Error: order must be [-99999; 99999] |
| 11 | Длина мантиссы более 30  во втором числе | 13e12 | | 2222222222222222222222222222222 | Error: mantissa cant be >30 symbols |
| 12 | Деление на ноль | 42 | | 0 | Error: zero division |
| 13 | Деление положительного на положительное | 0.2e1 | | 0.1e1 | +0.2e1 |
| 14 | Деление отрицательного на положительное | -1 | | 2 | -0.5e0 |
| 15 | Деление положительного на отрицательное | 9 | | -3 | -0.3e1 |
| 16 | Деление отрицательного на отрицательное | -3 | | -4 | +0.75e0 |
| 17 | Деление с результатом периодическая дробь | 1 | | 3 | +0.3333333333333333333333333333333e0 |
| 18 | Результат - периодическая дробь с округлением | 2 | | 3 | +0.666666666666666666666666666667e0 |
| 19 | Делимое намного больше делителя | 3000 | | 6 | +0.5e3 |
| 20 | Обычная проверка на порядок | 19e50 | | 19 | +0.1e51 |
| 21 | Вторая проверка на порядок | 1.9e50 | | 19e50 | +0.1e0 |
| 22 | Деление ноля | 0e12 | | 1.9 | +0.0e0 |
| 23 | Обычный тест | 0.5 | | 2 | +0.25e0 |
| 24 | Проверка порядка и мантиссы | 0.3333 | | 1111 | +0.3e-3 |
| 25 | Порядок результата на границе | 1e99999 | | 10 | 0.1e99999 |
| 26 | Порядок результата вышел за границу | 10e99999 | | 10 | Error: order was overflowed after division |
| 27 | Интересный тест | 100000000000000000000000000001 | | 900000000000000000000000000009 | +0.1111111111111111111111111111111e0 |
| 28 | Результат - округление вплоть до последней цифры | 999999999999999999999999999999 | | 2 | +0.5e30 |
| 29 | Неинтересный тест | 11 | | 13 | +0.846153846153846153846153846154e0 |
| 30 | Интересный тест | 123456789 | | 11 | +0.112233444545454545454545454545e8 |
| 31 | Пи(почти) | 104348 | | 33215 | +0.314159265392142144787159415927e1 |
| 32 | E идет сразу после точки | | 12.e1 | 2 | +0.6e2 |

**Контрольные вопросы**

1. **Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете?**

**Разрежённая матрица** — матрица с преимущественно нулевыми элементами. В противном случае, если большая часть элементов матрицы ненулевые, матрица считается ***плотной***.

**Словарь по ключам (DOK - Dictionary of Keys)** строится как словарь, где ключ — это пара (*строка, столбец*), а значение — это соответствующий строке и столбцу элемент матрицы.

**Список списков (LIL - List of Lists)** строится как список строк, где *строка* — это список узлов вида (*столбец, значение*).

**Список координат (COO - Coordinate list)** хранится список из элементов вида (строка, столбец, значение).

1. **Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы?**

Под хранение обычной матрицы выделяется sizeof(elem) \* n \* m байт – объем памяти возрастает квадратически.

Под хранение памяти выделяется

1. **Каков принцип обработки разреженной матрицы?**
2. **В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит?**

Зависит от плотности матрицы. В случае низкой разреженности матрицы стандартные алгоритмы обработки матриц более эффективны.

**Вывод**

В ходе выполнения данной лабораторной работы я научился реализовывать алгоритмы обработки разреженных матриц, включая их перевод в стандартную форму и обратно, сравнил эффективность использования алгоритмов (по времени выполнения и по требуемой памяти) со стандартными алгоритмами обработки матриц при различном процентном заполнении матриц ненулевыми значениями и при различных размерах матриц.

В 4444444