Склифасовский Денис ИУ7-35

Вариант 20

Лабораторная работа №1

«Длинная арифметика». Тип данных – массив.

**Цель работы:** реализация арифметических операций над числами, выходящими за разрядную сетку персонального компьютера, выбор необходимых типов данных для хранения и обработки указанных чисел.

**Условие задачи** (вар 20):

Составить программу умножения или деления двух чисел, где порядок имеет до 5 знаков: от –99999 до +99999, а мантисса – до 30 знаков. Программа должна осуществлять ввод чисел и выдавать либо верный результат в указанном формате (при корректных данных), либо сообщение о невозможности произвести счет.

Смоделировать операцию умножения действительного числа 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.

**Входные данные:**

Два действительных числа в форме ±m.n Е ±K.

**Выходные данные:**

Результат умножения в форме ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Возможные ошибки:**

1. Ошибка ввода: наличие непредусмотренных символов в поле ввода, пустое поле ввода, мантисса более 30 знаков, модуль порядка превосходит 99999;
2. Умножение на ноль;
3. Превышение допустимого порядка при обработке.

**Обращение к программе** осуществляется через консоль.

**Алгоритм:**

**1.** Ввод данных

**a)** Считывание

**b)** Проверка корректности

**c)** Нормализация

**2.** Умножение

Обнулить массив результата

Пока не конец первого числа

Пока не конец второго числа

Вставить произведение во временный массив

Конец пока

Пока не конец результата

К последующему элементу прибавить остаток от деления на 10

Элементу присвоить остаток от деления на 10

Конец ока

**3.**Вывод результата

а) Узнаем количество значащих чисел

б) Выводим значащие числа

в) Выводим порядок

**Структура данных:**

Числа хранятся в массиве типа char. Порядок, знак и кол-ва элементов массива хранятся в переменных типа int.

**Функции:**

Функция ввода:

*void* input\_float(*char* \*array, *int* \*sign, *int* \*count\_num, *int* \*point, *int* \*point\_pos, *int* \*exp, *int* \*err)

Ввод числа, определение его знака, размера, места точки и экспоненты

Функция нормализации:

*void* normalize(*char* \*array, *int* count, *int* point\_pos, *int* \*exp)

Преобразует массив, перенося все цифры после запятой

Функция проверки нулевого массива

*int* zero\_array(*char* \*array, *int* size)

Проверяет является ли массив нулевым

Функция преобразования массива

*void* mirror\_array(*char* \*array, *int* count)

«Переворачивает» массив для удобства работы с ним

Функция умножения

*void* multiplication(*char* \*first\_array, *char* \*second\_array, *int* first\_count, *int* second\_count, *int* \*c, *int* \*ind)

Перемножает числа

Функции округления

*int* round\_zero(*int* \*res, *int* ind)

Округляет незначимые нули

*int* round\_arr(*int* \*array, *int* count)

Округляет число, пока больше 30 цифр

*int* round\_exp(*int* \*array, *int* count, *int* \*exp)

Округление значимых нулей перед числом

**Тесты**

1. Корректный ввод

|  |  |  |
| --- | --- | --- |
| Вещественное | Вещественное | Результат |
| 0.5e-1 | 0.5 | +0.25e-1 |
| 0.00000003 | 3 | +0.9e-7 |
| 10. | 0 | +0. |

1. Граничные значения

|  |  |  |
| --- | --- | --- |
| Вещественное | Вещественное | Результат |
| +999999999999999999999999999999.e+99999 | 1 | Переполнение порядка |
| +999999999999999999999  999999999.e-99999 | +999999999999999999999999999999 | +0.999999999999999999999999999998e-99939 |
| +999999999999999999999999999999.e+100000 | 1 | Неверные входные данные |
| 1000000000000000000000000000000 | 1 | Слишком много цифр |
| 1e100000 | 1 | Экспонента должна быть [-99999; 99999] |

1. Ошибка ввода

|  |  |  |
| --- | --- | --- |
| Sample text | 1 | Неверный ввод |
| +-1 | 1 | Неверный ввод |
| 1 | -- | Неверный ввод |
| 1..1 | 1 | Неверный ввод |
| 1 e 0 | 1 | Неверный ввод |
| 1,5 | 1 | Неверный ввод |

**Контрольные вопросы:**

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

Диапазон значений чисел зависит от разрядности процессора, размера выделенной для хранения переменной памяти, от типа (целое/вещественное), знака. Под хранение целого положительного числа, для которого выделено 64 разряда, максимально возможное значение числа равно 2^64 – 1 = 18 446 744 073 709 551 615.

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

Точность представления вещественного числа зависит от максимально возможной длины мантиссы. Если длина мантиссы выходит за границы разрядной сетки, то происходит округление.

**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 специальные библиотеки для работы с большими числами; создать массив цифр числа и производить операции поэлементно, храня промежуточный результат в отдельном массиве; создать структуру, деля в ней мантиссу на части.

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

В данном случае возможно использовать поразрядные операции.

**Вывод:** При реализации арифметики длинных чисел можно хранить цифры числа, как элементы массива, производя операции над ними поразрядно.