Оберган Татьяна ИУ7-35Б

Вариант 18

**Лабораторная работа №1**

Обработка больших чисел

**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 1) :**

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 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 ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр

**Входные данные:**

Делимое: Действительное число в форме ±m.n Е ±K, суммарная длина мантиссы менее 30 цифр, величина порядка K - до 5 цифр;

Делитель: Целое число до 30 цифр.

**Выходные данные:**

Результат деления 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. Деление 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.** Деление

Проверяем степени двух чисел и, при необходимости, сдвигаем порядок.

До тех пор, пока кол-во полученных символов не равно 31 выполняем их нахождение:

Если делимое равно делителю, то текущий символ равен 1;

Если делимое меньше делителя, то производится сдвиг делителя на 1 позицию в лево (умножение на 10)

Если делимое больше делителя:

Пока делимое больше вычитаем делитель

счетчик++;

**3.** Вывод результата

а) Узнаем количество значащих чисел

б) Выводим значащие числа

в) Выводим порядок

**Структура данных:**

Длинные числа хранятся в массивах типа int. Где один элемент массива – цифра. Порядок хранится в отдельной переменной типа int. Знак хранится в переменной типа int (-1 – отрицательное, 1 – положительное).

**Функции:**

Функции ввода:  
В них заполняется массив, обрабатывается некорректность ввода

int input\_float(int \*m\_m, int \*size, int \*sign, int \*exponent);

int input\_int(int \*m\_i, int \*size, int \*sign, int \*exponent);

Функции нормализации:

Убирают лишние нули, сдвигают массив для дальнейшей обработки.

void normalize\_int(int \*arr, int \*size);

void normalize\_float(int \*arr, int \*size, int dot, int \*exponent);

Функция count\_zero:

Считает количество нулей в начале массива (необходимо при нормализации)

int count\_zero(int \*arr, int size);

Функция move\_digits:

Сдвигает все элементы массива на заданное (poryadok) расстояние

void move\_digits(int \*arr\_, int size, int poryadok);

Функция check\_ten:

Проверяет не переходит ли результат деления за 10.

void check\_ten(int \*arr\_float, int \*arr\_int, int size, int \*exp);

Функции деления:

Сравнивает два массива. Возвращает res; -1 Значит, что a < b

int comparison(int \*a, int \*b, int size);

Вычитает из массива a массив b

void subtract(int \*a, int \*b, int size);

Производит операцию деления двух массивов

void division(int \*arr\_float, int \*arr\_int, int \*result, int size, int \*exp);

Функция округления массива по полследнему символу:

void rounding(int \*arr, int size, int last\_digit);

Функция вывода результата:

void output\_res(int \*arr, int size, int exp, int sign);

**Тесты:**  
1) Корректный ввод

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Результат |
| -899e0 | -999 | +0.899899..e0 |
| 0.9e-4 | 81 | +0.111..e-5 |
| 1e5 | -17 | -0.588…e4 |

2) Ошибка ввода

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Результат |
| 1e0 | c | Wrong input |
|  | --- | Wrong input |
| 1e0 |  | Wrong input |

3) Деление на ноль

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Результат |
| 0.124E2 | 0 | Can’t divide by zero |

4) Граничные значения

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Результат |
| .99999..(30)e0 | 9999..(30) | +0.1e-29 |
| 99999..(30)e0 | 99999..(30) | +0.1e1 |
|  |  |  |

5) Нормализация -

\*Нельзя вводить делимое с незначащим последними нулями

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Частное |
| 035.126e0 | 1 | +0.35126e2 |
| 00.00351e0 | 1 | +0.351e-2 |
| 035.1260e0 | 1 | +0.35126e2 |
| 1000e0 | 1 | +0.1e4 |

6) Округление

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Частное |
| 1e0 | 3 | +0.3333....33e0 |
| -2e0 | 3 | -0.6666….67e0 |
| 9999..(30)e0 | 2 |  |

7) Превышение допустимого порядка

|  |  |  |
| --- | --- | --- |
| Делимое | Делитель | Частное |
| 1e100000(6) | --- | Exponent error |
| 1000..(31)e0 | --- | Input overflow |
| 1e0 | 1000..(31) | Input overflow |
| 5e99999(5) | 5 | Exponent error |
| 0.05e-99999(5) | 1 | Exponent error |

**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 ганицы 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 мантису на части.

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

В данном случае возможно использовать поразрядные операции.

**Вывод:**

При реализации аримфетики длинных чисел целесообразно хранить цифры числа, как элементы массива. Арифметику производить поразрядно.