

Лабораторная работа №1 по дисциплине «Типы и структуры данных»

«Длинная» арифметика. Тип данных – массив

Условие задачи:

Составить программу умножения двух чисел - целого, количество разрядов которого не больше 30, и вещественного, порядок которого находится в диапазоне от -99999 до +99999 (т.е. имеет не более 5 разрядов), а длина мантиисы не превышает 30 разрядов. Программа должна осуществлять ввод чисел в указанном диапазоне значений и выдавать результат в нормализованной форме $\pm 0.m1 E \pm K1$, где число $m1$ определено до 30 значащих цифр, число $K1$ – до 5 цифр. При невозможности произвести вычисления должно выдаваться соответствующее сообщение.

Исходные данные:

Целое число: число из цифр, количество которых не превышает 30, со знаком “+”, “-” или без знака (приравнивается к “+”) вещественное число: число со знаком (или без), состоящее из мантиисы, не превышающей 30 разрядов, и порядка со знаком (или без), не превышающего 5 цифр.

Формат входных данных:

Пользовательский ввод целого числа:

- +12345
- -12345
- 12345

Пользовательский ввод вещественного числа:

- 12.3E123 , -12.3E-123 , +12.3E+123 и т.д
- 123E123 // без точки
- .123E123
- 0.123E123
- 0.123e123
- 12.3 // без “E” или “e”

Вывод:

Формат вывода:

знак 0.мантииса E знак порядок

Примеры:

+0.12E-9999
-0.12E+9999

Возможные ошибки пользователя:

- 1) Введенное пользователем вещественное число, показатель степени E которого больше 9999 или менее -99999
Текст ошибки: "Ошибка в формате ввода."
- 2) Введенное пользователем вещественное число, показатель степени E которого при преобразовании становится больше 9999 или менее -99999
Текст ошибки: "Ошибка показателя E"
- 3) Ошибка при вычислениях из-за превышения показателем степени E допустимого предела 99999 или -99999
Текст ошибки: "Ошибка показателя E"
- 4) Некорректный ввод числа, содержащего лишние буквы или знаки
Текст ошибки: "Ошибка в формате ввода."
- 5) Некорректный ввод вещественного числа, содержащего лишние буквы или знаки
Текст ошибки: "Ошибка в формате ввода."

Алгоритм

1. Отображение описания программы
2. Считывание целого числа
3. Обработка строки ввода
 - a. Проверка введенного целого числа
 - b. Преобразование числа к необходимому виду
4. Считывание вещественного числа
5. Обработка строки ввода
 - a. Проверка введенного вещественного числа
 - b. Преобразование числа к необходимому виду
6. Выполнение умножения мантиссы
7. Обработка результатов умножения
 - a. Нахождение порядка
 - b. Округление
8. Печать результата

Структура данных

```
public struct number
{
    public bool sign;           //true означает +, false означает -
    public int[] mantissa;     // массив цифр для хранения мантиссы
    public int eps;            // показатель числа
}
```

Функции

Конвертирование строки в структуру

```
public static void Converter(string line, ref number num)
    string line -Строка считанная на вводе
    ref number num- указатель на структуру для хранения числа
```

Проверка ввода

```
public static int CheckInput(string line, string[] regarray)
    string line -Строка считанная на вводе
    string[] regarray -Массив шаблонов для проверки корректности
```

Печать мантиссы на экран

```
public static void mantisePrint(int [] mantiss)
    int [] mantiss -массив содержащий мантиссу
```

Умножение мантисс

```
public static void countmantiss(number num1, number num2, ref number
rez)
    number num1, number num2 -Структуры множителей
    ref number rez -Структура для хранения результата
```

Псевдокод функции:

pointer целое число

```
for i = num1.mantissa.Length-1 to 0 do
    pointer = 0;
    for j = num2.mantissa.Length-1 to 0 do
        rez.mantissa[i+j+1] += pointer + num1.mantissa[i]
        *num2.mantissa[j];
        pointer = rez.mantissa[i+j+1] / 10;
        rez.mantissa[i+j+1] %= 10;
    rez.mantissa [i] += pointer;
```

Округление

```
public static void round(ref number rez)
    ref number rez -Структура для хранения результата
```

Подсчет показателя

```
public static void couteps(number num1, number num2, ref number rez)
    number num1, number num2 -Структуры множителей
    ref number rez -Структура для хранения результата
```

Вычисление знака

```
public static void coutsign(number num1, number num2, ref number rez)
    number num1, number num2 -Структуры множителей
    ref number rez -Структура для хранения результата
```

Тесты

1) Стандартные входные значения

Целое число	Вещественное число	Результат
100	30.3E-3	+0.303E1
30	-.303	-0.909E1
5	-100.1E10	-0.5005E13

2) Границы порядка

Целое число	Вещественное число	Результат
1	0.01E-99999	0.01E-99998
1	0.1E99999	+0.1E99999
10	0.1E99999	Ошибка показателя E

3) Нормализация

Целое число	Вещественное число	Результат
1	0012.345	+0.12345E2
1	12.34500	+0.12345E2
1	0012.34500	+0.12345E2

4)Округление

Целое число	Вещественное число	Результат
999999999999999999 99999999999	2E0	+0.2E31
2	-777777777777777777 7777777777	-0.15555555555555555 5555555555555E31

5) Ошибочный ввод

Целое число	Вещественное число	Результат
-------------	--------------------	-----------

qqqqqq	qqqqqq	Ошибка в формате ввода.
25-26	1.1.1	Ошибка в формате ввода.
2E0	1.1E1.1+1	Ошибка в формате ввода.
1	qqqqqq	Ошибка в формате ввода.
1	1.1.1	Ошибка в формате ввода.
1	1.1E1.1+1	Ошибка в формате ввода.

6) Нулевой ввод

Целое число	Вещественное число	Результат
0	0.1E1	+0.0E0
10	0	+0.0E0
0	0	+0.0E0

Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон значений чисел зависит от размера области памяти, выделяемой под хранение переменной этого типа, от наличия знака в числе и от типа представления числа (целое или вещественное), от знака (signed). Диапазон чисел, представляемых в ПК зависит от разрядности процессора. Если процессор имеет 32 разряда, то максимальное значение составит $2^{32} - 1 = 4\,294\,967\,295$. Для 64 разрядов максимально возможное значение числа равно $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$.

2. Какова возможная точность представления чисел?

Точность представления вещественного числа зависит от максимально возможной длины мантиссы. При этом, если мантисса выходит за разрядную сетку ПК, то происходит ее округление. Для 64-разрядного процессора невозможно использовать больше 20 десятичных разрядов для

представления целого числа или более 20 знаков после точки в мантиссе для вещественного.

3. Какие стандартные операции возможны над числами?

Над числами возможно: сложение, вычитание, умножение, деление.

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

Наиболее предпочтительным форматом для хранения числа в памяти ПК является структура содержащая массив цифр мантиссы, значения показателя и общий знак числа. Данный тип хранения позволяет достаточно быстро и удобно проводить операции над числами.

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

Пользуясь стандартными алгоритмами арифметических операций, таких как умножение и деление в столбик.