Отчет

Условия задачи:

Составить программу умножения или деления двух чисел, порядок которых находится в диапазоне от –99999 до +99999 (т.е. имеет не более 5 разрядов), а длина мантиссы не превышает 30 разрядов.

Техническое задание:

Программа должна осуществлять ввод чисел формате ±m.n Е ±K, где суммарная длина мантиссы (m + n) до 30 значащих цифр, а величина порядка K до 5 цифр в указанном диапазоне значений и выдавать результат в нормализованной форме ±0.m1 Е ±K1. При невозможности произвести вычисления должно выдаваться соответствующее сообщение.

Описание работы с программой:

Запустить проект Xcode (расширение .xcodeproj). Скомпилировать и запустить программу выбрав в меню Product->Run. В нижним окне откроется консоль с инструкциями ввода.

Аварийные ситуации:

При зацикливание (зависании) следуют завершить работу программы выбрав в меню Product->Stop. При аварийном завершении работы Xcode, откройте проект заново.

Описание алгоритма:

Начало

А = Ввод первого числа

Б = Ввод второго числ а

Нормализовать А

Нормализовать Б

С = Умножение(А,Б)

Вывод С

Конец

Умножение (А,Б):

если A != 0 && Б != 0 то:

Результат.знак = !(А.знак ^ Б.знак)

если А.вектор.размер < Б.вектор.размер:

поменять(А,Б)

Результат.вектор.размер = А.вектор.размер + Б.вектор.размер

i = 0

пока i < А.вектор.размер:

i++

j = 0;

пока j < Б.вектор.размер:

j++

Результат.вектор[i+j] += А.вектор[i] \* Б.вектор[j]

Результат.порядок = А.порядок + Б.порядок

иначе:

Результат = 0

Описание структуры данных:

**typedef** vector<int> intVector;

**struct** longNum {

intVector vector; *// мантиса*

int degree = 0; *// порядок*

bool positive = true; *// знак числа*

bool flagZero = true; *// явяляется ли число нулем*

};

Тесты программы:

1. Стандартные данные:

1 \* 2 = +0.2E1

1000 \* 2 = +0.2E4

123 \* 10001 = +0.1230123 E7

135 \* 1.001 = +0.135135E3

10 \* 1000.142E12 = +0.1000142 E17

10 \* 1000.142E-12 = +0.1000142 E-7

-10 \* 2 = -0.2E2

10 \* (-2) = -0.2E2

1. Максимальные по вводу данные:

1 \* 99999 99999 99999 99999 99999 99999 E1 = +0.1E32

1 \* - 99999 99999 99999 99999 99999 99999 E1 = -0.1E32

2 \* 1.1E99999 = Ошибка – переполнение порядка

0.01 \* 1.1E-99999 = Ошибка – переполнение порядка

2 \* 1.1E-99999 = 0.22E-99998

999999999999999999999999999999 \* 1.0 = +0.1E31

-999999999999999999999999999999 \* 1.0 = -0.1E31

1. Максимальные по результату данные:

1 \* 1E99998 = 0.1E99999

0.1\* 1E-99999 = 0.1E-99999

1. Округление:

2 \* 666666666666666666666666666666 =0.1(3)E31

2 \* 888888888888888888888888888888 = 0.1(7)8E31

1. Тесты вызывающие ошибки ввода:

+-1

1.2323W20

24.2.4

35..4

1111111111111111111111111111111

1E100000

1E-100000

Ответы на вопросы:

1. Диапазон допустимых значений для целочисленного типа [-2(k-1)..(2(k-1)-1], k - длина числа в битах.
2. Для действительных чисел от 3.6E-4951 до 1.1E+4932, при длине мантиссы 52 разряда и 11 разрядов порядка
3. Стандартные операции над числами: сложение, вычитание, деление, умножение.
4. Программист создает свой тип (структуру данных) и реализовывает стандартные операции для него.
5. Длинные числа можно разбить на более простые и использовать метод вычисления “в столбик”.