Отчет по лабораторной работе № 1 по дисциплине

“Типы и структуры данных”

Выполнил: Студент ИУ7-31Б

Зейналов Зейнал Габибович

Работа № 1

«Длинная» арифметика. Тип данных – массив.

Цель работы: реализовать арифметические операции над числами, выходящими за разрядную сетку персонального компьютера, выбрать необходимые типы данных для хранения и обработки указанных чисел.

**описание условия задачи**

Смоделировать операцию умножения 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.

Программа должна осуществлять ввод чисел и выдавать либо верный результат в указанном формате (при корректных данных), либо сообщение о невозможности произвести счет.

**описание ТЗ**

1. Введение

**1.1** Наименование программы

Умножение большого вещественного и целого чисел до 30 знаков

**1.2** Краткая характеристика области применения

Данная программа позволяет умножать большие числа, размер которых выходит за разрядную сетку компьютера. Чаще всего, такие программы используются в системах спутникового  отслеживания и в астрономических расчетах.

**1.3** Сроки выполнения

Срок выполнения работы - 2 недели

**2.** Описание структур данных

**2.1** Заказчик

Заказчик данной работы является преподавательский состав по дисциплине “

Типы и структуры данных”

**2.2** Исполнитель

Исполнитель - Зейналов Зейнал Габибович, студент МГТУ им. Н.Э. Баумана, группы ИУ7-31б.

**2.3** Основание для разработки

Учебный процесс

3. Назначение разработки

3.1. Общая концепция системы

Программа предназначена для вычисления произведения вещественного и целого чисел больших разрядов. Пользователь вводит два числа и получает ответ в нормализованной форме.

3.2. Функциональность

Производит умножение вещественного и целого чисел, выходящих за рамки обычных типов.

4. Требования к программе

4.1. Требования к информационным структурам и методам решения

Данные должны хранится в массивах. Для решения необходимо выделить несколько осмысленных функций.

4.2. Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

* перемножать целое и вещественное числа, выходящие за разрядную сетку компьютера
* округлять ответ до 30 знаков

Программа должна осуществлять ввод чисел в указанном диапазоне значений и выдавать результат в нормализованной форме 0.m1 Е K1, где число m1 определено до 30 значащих цифр, число K1 – до 5 цифр. При невозможности произвести вычисления должно выдаваться соответствующее сообщение

Описание структур данных

В программе использованы следующие структуры данных:

b - массив элементов типа Char, хранит значение мантиссы первого элемента.

c - массив элементов типа Char, хранит значение мантиссы второго элемента.

r - массив элементов типа Int, хранит значение произведения двух мантисс.

por, count\_c, por - элементы типа Int, содержат значения порядков двух исходных чисел и полученного.

flag\_min,flag\_min1- элементы типа Char, содержат знаки двух исходных чисел.

count\_bef, count\_c, size - элементы типа Int, содержат длины массивов b,c,r.

Функции вывода результата на печать:

void show\_result(const char \*a, const int number)

void show\_result1(const int \*a, int number)

void show\_to\_user(const int \*result, const int por, int number, int flag)

Функция получения целого числа от пользователя:

int get\_from\_user1(char \*b, int \*number1, int \*flag)

Функция перемножения двух мантисс

int multiply(char \*a, char \*b, int number, int number1, int \*res)

Функция определения позиции сивола “Е” или “е”:

int pos\_e(char \*a, int \*pos, int \*count1, int \*flag)

Функция чтения мантиссы вещественного числа

int read\_mant(const char\* a, char\* b, int \*pos, int \*pos\_por, int \*flag\_min, int \*count\_bef, int \*flag1)

Функция чтения порядка вещественного числа:

int read\_por(const char \*a, int pos, int \*por)

Функция окргуления результата:

void round\_mant(int \*r, int \*count)

Функция замены строки:

int change\_str(char \*str, int count)

Описание алгоритма

Псевдокод функции умножения:

sum целое число

for i = num1.mantissa.Length-1 to 0 do

sum = 0;

for j = num2.mantissa.Length-1 to 0 do

rez.mantissa[i+j+1] += pointer + num1.mantissa[i] \*num2.mantissa[j];

sum = rez.mantissa[i+j+1] / 10;

rez.mantissa[i+j+1] %= 10;

rez.mantissa [i] += pointer;

Тесты

1) на работоспособность

|  |  |  |
| --- | --- | --- |
| Вещественное число | Целое число | Результат |
| 30.3Е-3 | 100 | +0.303Е1 |
| -.303 | 30 | -0.909E1 |
| -100.1E10 | 5 | -0.5005E13 |

2) на округление

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

3) на границах порядка

|  |  |  |
| --- | --- | --- |
| Вещественное число | Целое число | Результат |
| 0.01E-99999 | 1 | 0.01E-99998 |
| 0.1E99999 | 1 | +0.1E99999 |
| 0.1E99999 | 10 | Переполнение порядка |

4) Округление

|  |  |  |
| --- | --- | --- |
| Вещественное число | Целое число | Результат |
| 2E0 | 9999999999999999999 99999999999 | +0.2E31 |
| -7777777777777777777 77777777777 | 2 | -0.15555555555555555 5555555555555E31 |

5) неверный ввод

|  |  |  |
| --- | --- | --- |
| Вещественное число | Целое число | Результат |
| qqq |  | неккоректный ввод |
| 1 | йййй | неккоректный ввод |
| 1.1.1 |  | неккоректный ввод |

6) нули

|  |  |  |
| --- | --- | --- |
| Вещественное число | Целое число | Результат |
| 0 | 11111 | +0.0 E 0 |
| 2.1Е-2 | 0 | +0.0 E 0 |
| 0 | 0 | +0.0 E 0 |

Выводы по проделанной работе

В ходе проделанной работы создана программа, которая производит умножение двух вещественного чисел с высокой точностью. Студент изучил операции над числами больших разрядов и научился самостоятельно реализовывать действия над такими числами. Также была проработана тема строк в языке “СИ”.

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, сравнение.

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.