Лабораторная работа №1

**«Длинная» арифметика. Тип данных – массив**

**Выполнила: Лаврова Анастасия**

**Группа: ИУ7-35Б**

**Вариант №2**

**Условие 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.

**Техническое задание:**

1. Применение

«Длинная арифметика» необходима при создании ПО для процессоров низкой разрядности, а также для математического и финансового ПО.

Срок выполнения – 2 недели.

1. Входные данные

Входные данные разделены тремя полями: мантисса, порядок и целое число. Строка с мантиссой содержит число до 30 знаков, точку, допускается использование знаков «минус» и «плюс». Строка с порядком содержит число до 5 знаков, содержащее только цифры. Строка с целым числом содержит до 30 знаков, содержащее только цифры.

1. Выходные данные

Действительное число, представленное в форме 0.m Е K1, где m - до 30 значащих цифр, а K - до 5 цифр.

1. Описание задачи, реализуемой программой

Умножение действительного числа на целое. При достижении 31 и больше знаков мантиссы производится округление.

1. Способ обращения к программе

Запуск проекта *Arbitrary-precision arithmetic* в среде разработки QT Creator или Visual Studio. Можно использовать командную строку для запуска main.cpp

**Возможные аварийные ситуации**

**Некорректный ввод числа:**

Входные данные: последовательность знаков, содержащая иные символы помимо цифр, точки, знаков минус и плюс или же их неверная комбинация, а так же пустой ввод.

Выходные данные: сообщение о некорректном вводе в данном поле ввода

**Переполнение порядка в мантиссе:**

Входные данные: строка с более, чем с 30 значащих цифр мантиссы

Выходные данные: сообщение о переполнении мантиссы

**Переполнение порядка в степени:**

Входные данные: строка, более чем с 5 значащими цифрами степени

Выходные данные: сообщение о переполнении порядка

**Переполнение порядка в целом числе:**

Входные данные: строка, более чем с 30 значащих цифр

Выходные данные: сообщение о переполнение

**Переполнение порядка в степени в результате умножения:**

Входные данные: степень после произведения >99999

Выходные данные: сообщение о переполнении порядка степени

**Описание внутренних структур данных**

При считывании и проверки вводимых данных целесообразно использовать тип данных string. Для хранения в памяти и последующих операций с мантиссой используется динамический массив типа int.

string mnt;

int \*a = (int\*)calloc(mnt.length(), sizeof(int));

При считывании и проверки также использовался тип данных string. Далее недостающие нули добавляются в строку mnt.

string order;

При считывании и проверки также использовался тип данных string. Для хранения в памяти и последующих операций с целым числом используется динамический массив типа int.

string ing;

int \*b = (int\*)calloc(ing.length(), sizeof(int));

Для получения результата используется целочисленный динамический массив.

int \*c = (int\*)calloc(l, sizeof(int));

**Описание алгоритма:**

* Считывание строк и проверка ошибок ввода данных (функция check)

Если ошибок нет:

* Перевод строковых данных в целочисленные массивы
* Вычисление произведения (функция multiplication)
* Проверка на переполнение мантиссы

Если разряд переполнен:

* + Увеличение порядка степени
  + Округление мантиссы до 30 знаков
* Вывод ответа на экран

Алгоритм умножения:

Пока не кончились цифры в действительном числе:

Пока не кончились цифры в целом числе:

Перемножаем цифры в мантиссе и целом числе с конца и записываем в текущую ячейку ответа. (текущая ячейка ответа = индекс действительного числа + индекс целого числа - 1)

Пока не кончилась длина массива для ответа:

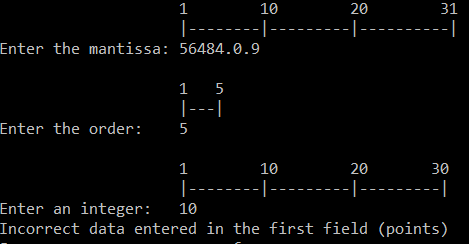
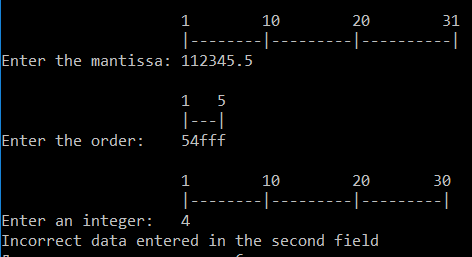
Перенос десятичного разряда на следующую цифру

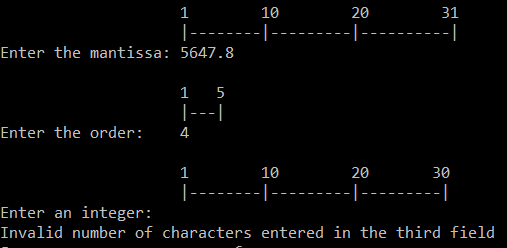
Пока не закончатся незначащие нули в начале массива:

Уменьшаем длину массива

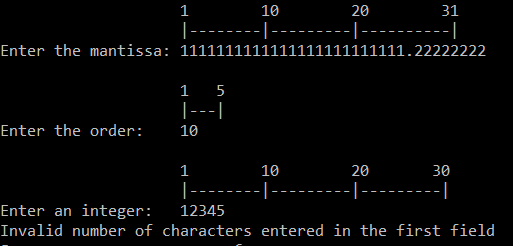
**Тесты**

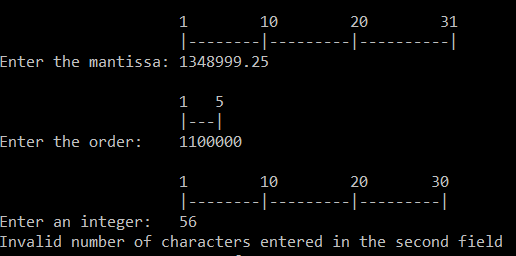
Некорректный ввод:

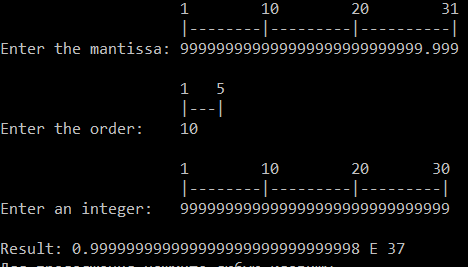
 

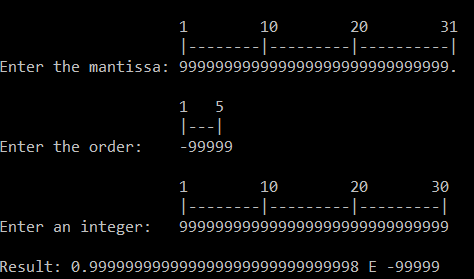
Пустой ввод:  


Переполнение при вводе:

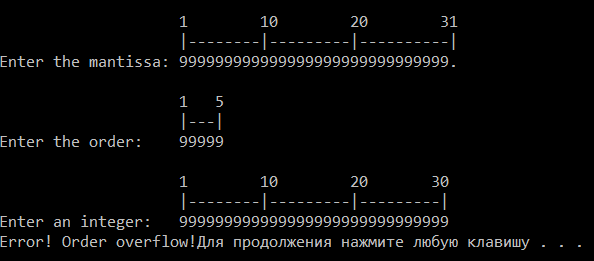


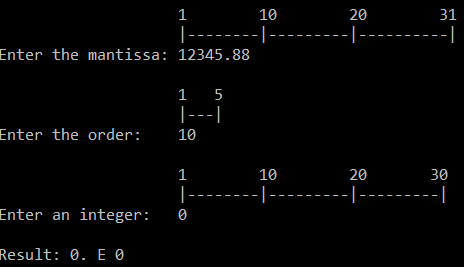


Проверка граничных значений:  




Переполнение степени:



Умножение на ноль:  


**Ответы на вопросы**

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от размера области памяти, выделяемой под хранение переменной , от наличия знака в числе и от типа представления числа (целое или вещественное). Для целочисленного типа от 0 до 2n-1 (беззнаковый), или от -2n-1 до 2n-1-1, если число со знаком, где n — число разрядов.

1. Какова возможная точность представления чисел, чем она определяется?

Вещественные числа хранятся в представлении с плавающей точкой в виде X = M \* E^p, где М – мантисса, представленная правильной дробью ( в интервале [0.1..1) ). Точность представления вещественного числа зависит от максимально возможной длины мантиссы. При этом, если мантисса выходит за разрядную сетку ПК, то происходит её округление. Для 64-разрядного числа на мантиссу отводится 53 разряда.

1. Какие стандартные операции возможны над числами?

Сложение, вычитание, умножение, деление.

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

Можно использовать массив из цифр – это является самым простым случаем представления целого беззнакового числа. Также можно использовать структуру, которая содержит мантиссу, порядок и флаг со знаком.

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

Путем последовательного выполнения операций над всеми цифрами в числе.