



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Московский государственный технический университет
имени Н.Э. Баумана»

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«ДЛИННАЯ АРИФМЕТИКА»

Студент

Фролов Евгений

Группа

ИУ7 – 35Б

2020 г.

Описания условия задачи

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

Смоделировать операцию умножения действительного числа на действительное число в форме $+/-m.n E +/-K$, где суммарная длина мантиссы $(m+n)$ - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $+/-0.m1 E +/-K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр.

Входные данные: Действительное число: строка, содержащая вещественное число в виде $<+/-m.nE+/-K>$. Знак перед числом и перед порядком обязательно нужно вводить. Также знак экспоненты $<e>$ нужно вводить обязательно и в верхнем регистре. Суммарная длина $<m+n>$ - до 31 цифры, включая точку; длина порядка — до 5 цифр. Пробелы при вводе числа недопустимы.

Можно: $+1.e+6$ $-35.3e+7$ $-132.e+5$;

Нельзя: $+1e+6$ $-35.$ $-132.e5$;

Выходные данные: «длинное число», нормализованное в виде $<+/-0.m1e+/-k1>$. Длина мантиссы $<m1>$ — до 30 цифр; длина порядка $<k1>$ — до 5 цифр.

Возможные аварийные ситуации:

- некорректный ввод (символы, пустой ввод, ввод без знака и т.д.);
- переполнение порядка в процессе умножения.

Алгоритм и структуры данных

Для записи вводимых данных используем массив символов. При успешном вводе действительного числа, записываем значащие цифры в целочисленный массив **int num_1[INT_MAX]**, где **INT_MAX** описан в **define** и равен 30, длина записывается в целочисленную переменную **int lenght_1**, а порядок в целочисленную переменную **int poryadok_1**, знак в целочисленную переменную **int sign_1** (-1, если знак отрицательный, 1 иначе), знак порядка в **int sign_1e**, если он есть (изначально **int sign_1e = -2**).

Аналогично для второго числа в целочисленный массив **int number_2[INT_MAX]**, длина массива(числа) в **int lenght_2**, а его порядок в **int poryadok_2**, знак в **int sign_2**, знак порядка в **int sign_2e**.

Результат умножения записывается в массив **int res[INT_MAX]**, длина массива в целочисленную переменную **int lenght**, а его порядок в целочисленную переменную **int poryadok**.

Знак числа и его порядка также записываются в отдельные целочисленные переменные **int sign**, **int sign_e**.

Основной алгоритм:

- Программа считывает действительное число в строку.

При успешном вводе: запись в массив значащих цифр и приглашение для ввода второго действительного числа. Иначе возвращаем код ошибки и сообщение. Аналогично для второго числа.

(int input_real(int *number_2,int *sign,int *lenght_2, int *poryadok, int *sign_e))

- После ввода двух действительных чисел, производим операцию умножения. Рассчитываем порядок результата и его знак. (Алгоритм ниже)

Если происходит переполнение порядка, то выдаем соответствующее сообщение и возвращаем код ошибки.

Если происходит переполнение мантиссы, округляем число до 30 знаков.

(void round_m(int *res, int *lenght))

Выдаем результат в нормализованном виде.

(void normalizacia(int sign, int *res, int lenght, int poryadok))

Алгоритм для умножения чисел:

```
(void multiply (int *a, int *b, int size_a, int size_b, int *res, int *lenght))
```

Функция для умножения чисел в столбик, в качестве аргумента принимает два числа в виде массива и их длины, указатель на массив и указатель на его длину. Функция ничего не возвращает.

Если мантисса числа a или $b = 0$, то в массив **int res[INT_MAX]** записываем 0.

Иначе находим максимально возможную длину результата и записываем в целочисленную переменную **int lenght** и заполняем массив **res** нулями.

Производим умножение чисел в «столбик». Числа передаются в массив в зеркальном виде.

Мы запускаем два цикла: один до **size_a**, другой до **size_b**. И в соответствующую ячейку записываем результат.

При умножении возможно может возникнуть переполнение, тогда делаем перенос разрядов. В каждой ячейке массива **res** должно располагаться однозначное число. Запускаем цикл до **lenght**. В рассматриваемое число записываем остаток от деления на 10, а к следующему добавляем целочисленное деление рассматриваемого числа на 10.

Результат также хранится в зеркальном виде.

Убираем лишние нули, если они есть. Они могут появиться, если длина результата оказалась меньше, чем изначально заданная (**size_a + size_b**) и узнаем точную длину результата умножения.

Тесты

[illegible]

Ответы к вопросам

1. Каков возможный диапазон чисел, представляемых в ПК?

Если под хранение целого положительного числа выделено 16 разрядов, то его максимальное значение не может превышать $2^{16}-1=65\,535$, если выделено 32 разряда, то максимальное значение составит $2^{32}-1=4\,294\,967\,295$. Для 64 разрядов максимально возможное значение числа равно $2^{64}-1=18\,446\,744\,073\,709\,551\,615$. Диапазон чисел зависит от выбранного типа, разрядности процессора и памяти выделенной для хранения числа.

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

Если десятичная точка расположена в мантиссе перед первой значащей цифрой числа, то при фиксированном количестве разрядов, отведённых под мантиссу, обеспечивается возможность сохранить максимальное количество значащих цифр, то есть обеспечить максимальную точность представления числа в ПК. Длина мантиссы определяет точность представления числа. При этом, если мантисса выходит за разрядную сетку ПК, то происходит ее округление. Для мантиссы числа типа double выделяется 52 бита, с помощью этого мантисса числа может иметь значение до $4\,503\,599\,627\,370\,496$.

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

Если обрабатываемые числа превышают возможный диапазон представления чисел в ПК, можно воспользоваться типами данных из дополнительно подключаемых библиотек, предназначенных для работы с большими числами или использовать массив для представления длинных чисел.

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

Осуществить эти операции можно, используя те же алгоритмы, которыми мы пользуемся, делая расчеты вручную или воспользоваться библиотечными функциями.

ВЫВОД

При написании лабораторной работы я познакомился с длинной арифметикой. Я понял, как располагаются числа в памяти компьютера и как происходит переполнение чисел. В процессе написания программы, научился обходить 10 ограничение языка программирования, создавая свои собственные операции для работы с такими числами.

В своей работе я реализовал возможность перемножения чисел, которые не умещаются в представлении компьютера. Алгоритм перемножения чисел реализован в виде умножения «в столбик».