



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 **«ДЛИННАЯ АРИФМЕТИКА»**

Студент Ковалец Кирилл Эдуардович

Группа ИУ7 – 33Б

Описание условия задачи

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

Если при умножении чисел длина мантиссы стала больше 30 знаков, то необходимо произвести округление (если 31-й разряд больше или равен 5, то к 30-му разряду добавляется единица, если меньше 5, то 31-й разряд отбрасывается).

Описание технического задания

Входные данные:

Два действительных числа, удовлетворяющие следующим правилам:

- 1) Числа вводятся только в экспоненциальной форме;
- 2) Оба числа вводятся со знаком (+) или (-);
- 3) Порядок числа вводится со знаком (+) или (-);
- 4) Длина мантиссы не должна превышать 30 цифр;
- 5) Величина порядка не должна превышать 5 цифр;
- 6) Латинская буква E должна быть заглавной;
- 7) В числе должна быть точка;
- 8) В записи числа не должно быть пробелов;
- 9) Каждое число вводится на отдельной строке;
- 10) Перед и после точки должна идти цифра.

Выходные данные:

Число в форме $\pm 0.m1 \text{ E } \pm K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр.

Действие программы:

Моделирование операции умножения действительного числа на действительное число.

Способ обращения к программе:

Запускается через терминал.

(с помощью команды `./Documents/Тисд/lab_1/lab_1/app.exe`)

Сообщения при аварийных ситуациях:

- 1) В записи числа содержится недопустимый символ;
- 2) Введена пустая строка;
- 3) Слишком длинная запись числа;

- 4) В записи числа должна быть одна точка;
- 5) В записи числа должны быть два знака;
- 6) В записи числа должна быть одна буква E;
- 7) Число должно начинаться с знака;
- 8) Ошибка в мантиссе числа. Встречен недопустимый символ;
- 9) Суммарная длина мантиссы должна иметь от 2 до 30 цифр;
- 10) Точка находится не на своём месте, проверьте правильность ввода мантиссы;
- 11) Перед величиной порядка должен идти знак;
- 12) Величина порядка должна иметь от 1 до 5 цифр;
- 13) Произошло переполнение.

Описание структуры данных

В данной лабораторной работе для хранения данных я использую массив символов длиной 39 (Это минимальное кол-во символов для хранения максимально возможного числа) и переменную типа `int` для хранения порядка числа.

После ввода число записывается в подготовленный для него массив, где проверяется на соответствие правилам ввода данных. Далее считывается порядок числа и записывается в переменную типа `int`. После ячейки массива, содержащей букву “E” идёт ячейка, содержащая знак порядка числа. Первый элемент массива содержит знак самого числа. Между знаками хранятся символы мантиссы.

`char string_1[MAXIMUM_STRING_LENGTH + 1]` – массив символов для хранения первого числа;

`int position_E1` – переменная для хранения позиции символа “E” в первом числе;

`int order_1`– переменная для хранения порядка первого числа;

`char string_2[MAXIMUM_STRING_LENGTH + 1]` – массив символов для хранения второго числа;

`int position_E2` – переменная для хранения позиции символа “E” во втором числе;

`int order_2`– переменная для хранения порядка второго числа;

`char result[MAXIMUM_LENGTH_OF_MANTISA * 2 + 1]` – массив символов для хранения результата перемножения мантисс двух чисел;

`int order` – переменная для хранения порядка результата вычисления.

```
#define MAXIMUM_STRING_LENGTH 39
#define MAXIMUM_LENGTH_OF_MANTISA 30
```

Описание алгоритма

Программа по очереди получает на вход две строки, содержащие действительные числа в экспоненциальной форме. Каждая строка проходит два этапа проверки. На первом этапе проверяются символы, входящие в строку. После такой проверки мы с уверенностью можем сказать, что в данной записи отсутствуют посторонние символы, что в числе содержится только одна точка, только одна заглавная латинская буква “Е” и два знака (“+” или “-”). Второй этап нужен для того, чтобы убедиться в правильности формата введенного числа.

После проверки мы можем перейти к подготовке к умножению. Для этого переведем часть строки (символы, стоящие после “Е”) в число, которое будет отвечать за порядок введенного числа. После этого присвоим элементу, стоящему за “Е”, символ конца строки для того, чтобы дальше работать только с мантиссой. Уберем из неё все незначащие нули, после чего нормализуем мантиссу.

Проделав подготовительную работу для двух чисел, приступаем к их умножению. Для этого создадим массив с размером под максимально возможную длину мантиссы нового числа, полученного после умножения. Заполняем массив нулями. Посимвольно перемножаем цифры мантисс двух чисел, прибавляя промежуточные результаты вычисления элементам массива. Избавимся от незначащих нулей. В итоге получим массив из цифр новой мантиссы. Для получения порядка нового числа сложим порядки двух множителей. Проверим новое число на переполнение. В случае, если мантисса получилась равной нулю, присвоим порядку так же ноль. При необходимости округлим мантиссу до 30 знаков после запятой.

В правильном формате выведем результаты вычисления.

Набор тестов

№	Что проверяется	Число №1	Число №2	Вывод
1	Наличие недопустимых символов	+123.02a-023	_____	В записи числа содержится недопустимый символ! "
2	Ввод пустой строки	\n	_____	Введена пустая строка!
3	Длина введенного числа	+11111111112222222 222.333333333444444 444E+11111111	_____	Слишком длинная запись числа!

4	Наличие точки в числе (по правилам ввода число не может не содержать дробную часть)	+12E-23	_____	В записи числа должна быть одна точка! (по правилам ввода число не может не содержать дробную часть)
5	Наличие двух знаков в числе (по правилам ввода число и его порядок должны начинаться со знака)	12.1E+2	_____	В записи числа должны быть два знака! (по правилам ввода число и его порядок должны начинаться со знака)
6	В записи числа одна буква 'Е'? (по правилам ввода число вводится только в экспоненциальной форме)	+12.1EE+2	_____	В записи числа должна быть одна буква Е! (по правилам ввода число вводится только в экспоненциальной форме)
7	Первый символ – знак?	12.23E++2	_____	Число должно начинаться со знака!
8	Все символы мантиссы, кроме точки – цифры?	+0.88-E2	_____	Ошибка в мантиссе числа. Встречен недопустимый символ '-'
9	Длина мантиссы превышает 30 цифр?	+0.123456789012345678901234567890E+2	_____	Суммарная длина мантиссы должна иметь от 2 до 30 цифр!
10	Наличие после точки цифры (по правилам ввода мантисса не может не иметь целую или дробную части)	+1.E-2	_____	Точка находится не на своём месте. (по правилам ввода мантисса не может не иметь целую или дробную части)
11	Наличие перед точкой цифры (по правилам ввода мантисса не может не иметь целую или дробную части)	+.999E+2	_____	Точка находится не на своём месте. (по правилам ввода мантисса не может не иметь целую или дробную части)

12	Знак перед порядком числа	+1.23E9+	_____	Перед величиной порядка должен идти знак!
13	Введён ли порядок?	+1.23E+	_____	Величина порядка должна иметь от 1 до 5 цифр!
14	Величина порядка должна иметь не больше 5 цифр	+1.23E+111111	_____	Величина порядка должна иметь от 1 до 5 цифр!
15	Порядок нового числа не должен превышать 99999	+9.9E+99999	+9.9E+99999	Произошло переполнение!
16	Порядок нового числа должен быть больше -99999	+9.9E-99999	+9.9E-99999	Произошло переполнение!
17	Умножение положительного числа на отрицательное	+12.2E+1	-3.0E+1	-0.366 E +4
18	Умножение чисел, порядки которых имеют разные знаки	+2.2E+10	+3.0E-20	+0.66 E -9
19	Порядок одного из двух чисел после нормализации превышает 99999	+999999999999999999 999999999999.9E+99 999	+1.0E-100	+0.9999999999999999 9999999999999999 E +99928
20	Умножение на ноль	-9.12E+5	+0.0E+0	+0.0 E +0
21	Умножение без округления (31 цифра мантиссы < 5)	+999999999999999999 999999999999.9E+10	+9999999999 999999999999 999999999.9E +10	+0.9999999999999999 9999999999999998 E +78
22	Умножение с округлением результата «паровозиком»	+999999999999999999 999999999999.9E+10	+5.0E+0	+0.5 E +40
23	Умножение с округлением результата	+999999999999999999 999999999999.9E+10	+32.0E+0	+0.3199999999999999 9999999999999997 E +40
24	Обычный пример	+999.9E+12	+32.0E+0	+0.319968 E +17

Функции

INT STRING_INPUT_WITH_VALIDATION(CHAR *S, CONST INT N);

Функция ввода числа с первичной проверкой.

Аргументы:

- Указатель на строку, куда будет записано число;
- Максимальная длина строки.

Возвращает код ошибки.

Изменяет:

- Строку, содержащую число;

Возможные коды ошибок:

- INVALID_CHARACTER;
- TOO_LONG_NUMBER_ENTER;
- EMPTY_LINE;
- THERE_MUST_BE_ONE_POINT;
- THERE_MUST_BE_TWO_SIGNS;
- THERE_SHOULD_BE_ONE_LETTER_E;
- OK.

INT STRING_VALIDATION(CONST CHAR *STR, INT *POSITION_E);

Функция осуществляет оставшуюся проверку числа.

Аргументы:

- Указатель на строку с числом;
- Указатель на элемент, в который будет записана позиция символа "E".

Возвращает код ошибки.

Изменяет:

- Элемент, содержащий позицию числа "E".

Возможные коды ошибок:

- NUMBER_MUST_START_WITH_A_SIGN;
- ERROR_IN_MANTISSA_NUMBER;
- WRONG_LENGTH_OF_MANTISSA;
- POINT_OUT_OF_PLACE;
- NO_SIGN_IN_FRONT_OF_THE_ORDER_VALUE;
- WRONG_ORDER_LENGTH;
- OK.

INT CONVERT_STRING_TO_NUMBER(CONST CHAR *STRING);

Функция переводит строку в число.

Аргументы:

- Указатель на строку.

Возвращает число.

Функция ничего не изменяет.

INT NUMBER_NORMALIZATION(CHAR *STRING, CONST INT POSITION_E);

Функция нормализует число.

Аргументы:

- Указатель на строку, содержащую число;
- Указатель на элемент, содержащий позицию символа "Е".

Возвращает изменение порядка числа;

Изменяет:

- Строку с числом.

VOID FILL_WITH_ZEROS(CHAR *RESULT, CONST INT N);

Функция заполняет массив символом '0'.

Аргументы:

- Указатель на массив;
- Количество элементов массива.

Функция ничего не возвращает.

Изменяет:

- Массив, переданный по указателю.

VOID ROUNDING(CHAR *RESULT);

Функция округляет число, записанное в строке, при необходимости.

Аргументы:

- Указатель на строку, содержащую число;

Функция ничего не возвращает.

Изменяет:

- Строку, переданную по указателю.


```
int multiplication(char *result, int *order, const char
*string_1, const char *string_2, const int order_1,
const int order_2);
```

Функция умножения двух чисел.

Аргументы:

- Указатель на строку содержащую первое число;
- Указатель на строку содержащую второе число;
- Указатель на строку, содержащую результат вычисления;
- Порядок первого числа;
- Порядок второго числа;
- Указатель на порядок результата вычисления.

Возвращает код ошибки.

Изменяет:

- Строку, содержащую результат вычисления;
- Порядок результат вычисления.

Возможные коды ошибок:

- OVERFLOW_OCCURRED;
- OK.

Вывод

В данной лабораторной работе я использовал массив символов для хранения данных (знака числа, мантиссы, знака порядка и самого порядка), что оказалось весьма неудобным. Получив опыт работы с массивом, я сделал выводы и в следующий раз для похожих задач буду использовать структуры и объединения.

Также в процессе выполнения данной лабораторной работы я научился обрабатывать числа, выходящие за рамки машинного представления, складывать и умножать их. Наглядно убедился, что и у компьютера есть предел возможностей, а также познакомился с представлением числа в его памяти.

Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон чисел зависит от выбранного типа, разрядности процессора и памяти выделенной для хранения числа. Максимальное значение беззаконного целого числа, для которого выделяется 64 разряда, равно 2^{64} (18 446 744 073 709 551 615).

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

Точность представления вещественных чисел определяется количеством памяти, выделяемой для хранения мантиссы числа. Для мантиссы числа типа double выделяется 52 бита, с помощью этого мантисса числа может иметь значение до 2^{52} (4 503 599 627 370 496).

3. Какие стандартные операции возможны над числами?

Возможны операции сложения, вычитания, умножения, деление, взятие остатка, сравнение.

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

Программист может использовать массив символов или выбрать структуру, куда запишет мантиссу, порядок, знак числа, знак порядка.

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

Можно использовать собственные функции или найти подходящие в библиотеках.