|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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. ТИП ДАННЫХ - МАССИВ**

Студент Ву Минь Куанг

Группа ИУ7И – 34Б

Преподаватель Силантьева А. В.

*2021 г.*

**Оглавление**

[**1.** **Описание условия задачи** 3](#_Toc82092627)

[**2.** **Описание ТЗ** 3](#_Toc82092628)

[**3.** **Описание внутренних структур данных** 4](#_Toc82092629)

[**4.** **Описание алгоритма** 4](#_Toc82092630)

[**5.** **Набор тестов с указанием, что проверяется** 6](#_Toc82092631)

[**6.** **Выводы по проделанной работе** 7](#_Toc82092632)

[**7.** **Ответы на вопросы** 7](#_Toc82092633)

# **Описание условия задачи**

Составить программу умножения двух чисел, где порядок имеет до 5 знаков: от –99999 до +99999, а мантисса – до 30 знаков. Программа должна осуществлять ввод чисел и выдавать либо верный результат в указанном формате (при корректных данных), либо сообщение о невозможности произвести счет.

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

# **Описание ТЗ**

1. **Исходные данные**: 2 строки, содержащие два действительных числа вида: [+/-]m.nE[+/-]k, где:
   * Суммарная длина мантиссы m+n – до 30 цифр.
   * Порядка k – до 5 цифр.
   * Допускается отсутствие точек, e, знаков.
2. **Результат**: действительное число в форме 0.m1 е k1, где:
   * m1 - до 30 значащих цифр.
   * k1 - до 5 цифр.
3. **Задачи, реализуемой программой**: произведение действительного числа на действительное число.

**Аварийные ситуации:**

1. Некорректный ввод: строка с вещественным числом содержит символ не цифру и не «.».
2. Некорректный ввод: строка с вещественным числом содержит более одной точки.
3. Некорректный ввод: превышение длины при вводе вещественного числа (больше 31 цифры, включая точку)
4. Некорректный ввод: введена только точка
5. Некорректный ввод: введена пустая строка (т.е. «\n»).
6. Некорректный вывод: в порядке введено не число
7. Некорректный вывод: порядок превышает число «99999»
8. Некорректный вывод: переполнение порядка при умножении

# **Описание внутренних структур данных**

Структурой данных является структура, содержащая:

1. Знак мантиссы (char)
2. Мантисса числа (int)
3. Знак порядка (char)
4. Порядок (е) (int)

// Структура, содержащая характеристики числа

#define MAX\_MANTISSA\_LEN 32

typedef struct{

char mantissa\_sign;

int mantissa\_int[MAX\_MANTISSA\_LEN];

int mantissa\_int\_len;

int mantissa\_float[MAX\_MANTISSA\_LEN];

int mantissa\_float\_len;

char power\_sign;

int power;

} real\_number;

# **Описание алгоритма**

1. Программа считывает две строки, которые содержат вещественные числа, и записывает части числа в нужные места структуры *real\_number*

**(int read\_number(real\_number \*number))**

1. Далее правильно введенные числа передаются в функцию нормализации, которая убирает ненужные нули и сдвигает точку в начало числа, изменяя его порядок

**(int normalization(real\_number \*number))**

1. После успешно проведенной нормализации числа передаются в функцию умножения, в которой создается массив максимально возможной длины числа с элементами типа *int* и в него по принципу умножения «в столбик» записываются числа, учитывая все переносы десятков в новые разряды числа. После умножения в числе проводится проверка, которая при необходимости, удаляет лишние нули или округляет число. Затем результат выводится в нормализованном виде в соответствии со спецификацией, указанной в ТЗ (<+\-0.m1E+\-K1>)

**(*int real\_mult(real\_number \*a, real\_number \*b, real\_number \*res)*)**

**Функции программы**

1. ***int read\_number(real\_number \*number)***
2. **Описание**: функция совершает чтение вещественного числа и его последующую запись в необходимые поля структуры *number*
3. **Входные значения**: структура *number* для записи в нее числа
4. **Выходные значения**: структура *number* с записанным в нее числом; *rc* – код ошибки (или ноль при ее отсутствии)
5. ***int normalization(real\_number \*number)***
6. **Описание**: функция нормализует мантиссу числа, записанную в *number*, и меняет порядок числа, а также удаляет лишние нули
7. **Входные значения**: структура *number* для нормализации числа
8. **Выходные значения**: структура *number* с нормализованным в ней числом
9. ***int real\_mult(real\_number \*a, real\_number \*b, real\_number \*res )***
10. **Описание**: Функция перемножает два вещественных числа
11. **Входные значения**: структуры num1 и num2, числа которых необходимо перемножить; массив *mantissa\_float*, в который будет записан результат вычислений; *power* – порядок получившегося числа; *(MANTISSA\_LEN - i)* – в каком месте массива *mantissa\_float* находится получившееся число
12. **Выходные значения**: массив *mantissa\_float*, в который будет записан результат вычислений; *power* – порядок получившегося числа; *(MANTISSA\_LEN - i)* – в каком месте массива *mantissa\_float* находится получившееся число; *rc* – код ошибки (или ноль при его отсутствии)

# **Набор тестов с указанием, что проверяется**

Тепличные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Число №1 | Число №2 | Выходные данные | Проверяется |
| 1 | 2.0 | 6.0 | +0.12е+2 | Корректность умножения |
| 2 | -2.0 | -6.0 | +0.12е+2 | Корректность умножения знаков - на - |
| 3 | 2.0 | -6.0 | -0.12е+2 | Корректность умножения знаков + на - |
| 4 | -2.0 | 6.0 | -0.12е+2 | Корректность умножения знаков - на + |
| 5 | 1.2e2 | 1.2e2 | +0.144e+5 | Корректность умножения экспонециальных чисел |
| 6 | 1.2e2 | 120 | +0.144e+5 | Корректность умножения экспон. на веществен. |
| 7 | 123.123 | 123.123 | +0.12159273129е+5 | Умножения больших чисел |
| 8 | 99...99(30) | 2 | 0,2е+31 | Округление |
| 9 | 999е99999 | 1 |  | Переполнение порядка |
| 10 | 0.001е-99999 | 1 |  | Переполнение порядка |

Граничные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Число №1 | Число №2 | Выходные данные | Проверяется |
| 1 | 0.00000 | 0.00000 | +0.0е+0 | Нулевое значение в вещественном формате |

Запредельные:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Число №1 | Число №2 | Выходные данные |
| 1 | 1231ewe21 | - | Некорректные символы! |
| 2 | 123e.21 | - | 'e' идёт после '.' |
| 3 | -1221.e-12-12 | - | Слишком много знаков '-' |
| 4 | dfdwe | - | Некорректные символы! |
| 5 | 0.1e123456 | - | Слишком большой порядок |
| 6 | 0.1e-123456 | - | Слишком большой порядок |
| 7 | 0.999999999999998000000000000001e12345 | - | Слишком большая мантисса |
| 8 | 0.0.0.0 | - | "." встретилась более 1 раза! |
| 9 | 0.2e33e3 | - | "E" встретилась более 1 раза! |

# **Выводы по проделанной работе**

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».

# **Ответы на вопросы**

1. **Каков возможный диапазон чисел, представляемых в ПК?**

Диапазон чисел зависит от наличия/отсутствия знака у числа, выбранного типа, отведенной для него памяти, разрядности процессора. Например: знакового короткого целого, под который выделено 2 байта диапазон будет - [−32768, +32767].

Максимально под представление мантиссы отводится 52 двоичных разряда, а под представление порядка – 11 разрядов. В этом случае возможные значения чисел находятся в диапазоне от 3.6 E –4951 до 1.1 E +4932.

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

Точность вещественного числа определяется количеством разрядов, отведённых для хранения мантиссы. Например: из 4 байт, выделенных под хранение цифры, 1 (8 разрядов) отдается под данные о порядке и его знаке, а 3 байта (24 разряда) уходят на хранение мантиссы и её знака по тем же принципам, что и для целочисленных значений. Тогда точность будет равна 7 десятичным знакам.

Максимально под представление мантиссы отводится 52 двоичных разряда, а под представление порядка – 11 двоичных разрядов. В этом случае возможные значения чисел находятся в диапазоне от 3.6 E –4951 до 1.1 E +4932.

1. **Какие стандартные операции возможны над числами?**

Сложение, вычитание, умножение, деление, сравнение.

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

Для представления такого числа программист может разбить число на мантиссу и порядок, на числа/цифры (с помощью массива чисел/символов/строк), отдельно хранить знаки мантиссы и порядка. Каждый из этих элементов должен удовлетворять диапазону представления чисел в ПК.

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

Можно реализовать собственные функции, которые выполняли бы указанные операции, используя представление числа в заранее выбранном формате (см пункт 4).

baryshnikovam@mail.ru