|  |  |
| --- | --- |
| 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» (ИУ) |
| Кафедра | «Программное обеспечение ЭВМ и информационные технологии» (ИУ7) |

**Лабораторная работа №1**

**“Обработка больших чисел”**

**Вариант №8**

|  |  |
| --- | --- |
| Студент: Князев Дмитрий Юрьевич, группа ИУ7-33Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |
| Преподаватель:  Силантьева Александра Васильевна | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Оглавление

Условия задачи 2

Структура данных и описание функций 3

Описание алгоритма 4

Ограничения и функциональные тесты 10

Вывод 13

Контрольные вопросы 14

**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

**Входные данные:**

Первый множитель: Действительное число в форме ±m.n Е ±K, суммарная длина мантиссы менее 30 цифр, величина порядка K - до 5 цифр;

Второй множитель: Действительное число в форме ±m.n Е ±K, суммарная длина мантиссы менее 30 цифр, величина порядка K - до 5 цифр;

**Выходные данные:**

Результат умножения в форме ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Структура данных:**

Структура данных представляет собой совокупность полей порядка (int32\_t) и большого длинного числа, способного расширяться динамически (структура longint\_t), которое включает в себя массив элементов (типа uint32\_t), представляющих разряды числа в системе счисления с основанием 2^30, и размер этого массива (1 объект типа int32\_t).

**Описание функций:**

|  |
| --- |
| **typedef** **struct**  {  sdigit exponent;  **longint\_t** \*mantissa;  } **bigfloat\_t**; |

Структура большого числа с плавающей точкой.

В качестве мантиссы выступает динамически расширяемый тип longint\_t (большое целое число), а в качестве порядка — sdigit (int32\_t)

|  |
| --- |
| **bigfloat\_t** \*  **new\_bigfloat**(**void**); |

Динамически выделяет память под хранение нового объекта с помощью функции malloc, присваивает полю mantissa значение NULL, полю exponent — значение 0.

Принимает: ничего

Возвращает: в случае успешного выделения памяти - указатель на новое большое число с плавающей точкой, иначе - NULL

|  |
| --- |
| **void**  **free\_bigfloat**(**bigfloat\_t** \*\*z); |

Освобождает память, выделенную функцией new\_bigfloat, освобождает память, выделенную для поля mantissa, если указатель отличен от NULL, присваивает NULL указателю, на который ссылается аргумент.

Принимает: указатель на bigfloat\_t

Возвращает: ничего

|  |
| --- |
| **bigfloat\_t** \*  **bigfloat\_mul**(**bigfloat\_t** \*a,  **bigfloat\_t** \*b); |

Производит умножение двух больших чисел с плавающей точкой.

Принимает: два указателя на объекты типа bigfloat\_t, выступающие в качестве множителей.

Возвращает: указатель на новый объект, являющийся округлённым результатом умножения двух чисел с плавающей точкой.

|  |
| --- |
| **void**  **print\_bigfloat**(**bigfloat\_t** \*v); |

Выводит в стандартный поток вывода большое число с плавающей точкой в форме ±0.ME±K

Принимает: указатель на объект типа bigfloat\_t

Возвращает: ничего

|  |
| --- |
| **void**  **set\_bigfloat**(**bigfloat\_t** \*z,  **longint\_t** \*new\_mantissa,  sdigit new\_exponent); |

Присваивает большому числу с плавающей точкой новой мантиссы и порядка, старое значение мантиссы при этом освобождается, если указывает не на NULL

Принимает: указатель на объект типа bigfloat\_t, указатель на новую мантиссу, новое значение порядка

Возвращает: ничего

|  |
| --- |
| **int**  **input\_bigfloat**(**bigfloat\_t** \*v); |

За один проход и фиксированным количеством памяти осуществляет ввод большого числа с плавающей точкой и присвоение аргументу указателя на новое округлённое большое число с плавающей точкой.

Принимает: указатель на тип bigfloat\_t

Возвращает: EXIT\_SUCCESS при успешном вводе

EXIT\_FAILURE при ошибке

|  |
| --- |
| **int**  round\_mantissa(**longint\_t** \*\*v) |

Округляет мантиссу путём преобразования типа longint\_t в строку десятичных цифр и наоборот

Принимает: указатель на указатель на тип bigfloat\_t

Возвращает: 0 при отсутствии переполнения при округлении

1 при возникновении переполнения при округлении

|  |
| --- |
| **typedef** **struct**  {  digit \*digits;  **signed\_size\_t** size;  } **longint\_t**; |

Cтруктура большого целого числа.

Большое число с плавающей точкой состоит из указателя на массив digit, хранящий число в системе счисления с основанием 2^30, и поля size, являющегося знаковой версией типа size\_t, показывающей размер массива.

|  |
| --- |
| **longint\_t** \*  **new\_longint**(**signed\_size\_t** size); |

Возвращает большое целое число с заданным размером массива разрядов

Принимает: начальный размер массива разрядов

Возвращает: указатель на новый объект типа longint\_t

|  |
| --- |
| **void**  **free\_longint**(**longint\_t** \*\*z); |

Освобождает память, выделенную для большое целое число, присваивает аргументу значение NULL

Принимает: указатель на указатель на большое целое число

Возвращает: ничего

|  |
| --- |
| **void**  **longint\_normalize**(**longint\_t** \*v); |

Удаляет незначащие нулевые элементы-разряды

Принимает: указатель на большое целое число

Возвращает: ничего

|  |
| --- |
| **longint\_t** \*  **longint\_from\_string**(**const** **char** \*str); |

Переводит строку в большое целое число

Принимает: указатель на строку

Возвращает: указатель на новый объект типа longint\_t

|  |
| --- |
| **char** \*  **longint\_to\_string**(**longint\_t** \*a); |

Переводит большое целое число в строку

Принимает: указатель на объект longint\_t

Возвращает: указатель на созданную строку в случае успеха, иначе — NULL

|  |
| --- |
| **int**  **get\_decimal\_longint\_length**(**longint\_t** \*a,  **size\_t** \*digit\_num); |

Считает количество десятичных разрядов в большим целом числе и присваивает результат второму аргументу

Принимает: указатель на большое целое число, указатель на количество десятичных разрядов

|  |
| --- |
| **longint\_t** \*  **long\_long\_mul**(**longint\_t** \*a,  **longint\_t** \*b); |

Умножает большие целые числа

Принимает: 2 указателя на объекты типа longint\_t, являющиеся множителями

Возвращает: указатель на новое большое целое число, являющееся результатом произведения

|  |
| --- |
| **int**  longint\_digit\_add(**longint\_t** \*\*a,  digit b) |

Прибавляет к большому целому числу a значение числа b типа digit.

Принимает: указатель на объект типа longint\_t, число типа digit

Возвращает: EXIT\_SUCCESS при успешном сложении

EXIT\_FAILURE при ошибке

|  |
| --- |
| **void**  set\_longint\_to\_digit(**longint\_t** \*z,  digit number) |

Присваивает большому целому числу значение number, присваивает полю size объекта z значение 1.

Принимает: указатель на объект типа longint\_t, цифру типа digit

Возвращает: ничего

|  |
| --- |
| **int**  **is\_longint\_zero**(**longint\_t** \*z); |

Проверяет, равно ли большое целое число нулю

Принимает: указатель на объект типа longint\_t

Возвращает: 1 в случае успеха, 0 - иначе

**Описание алгоритма:**

**1. Ввод данных**

1.1 Считываем до тех пор, пока кол-во полученных символов не больше 31.

1.2 Округляем число с возможным прибавлением единицы к порядку (П.3)

**2. Умножение**

2.1 Умножаем мантиссы в системе счисления 2^30

2.2 Складываем порядки

2.3 Проверка: если количество цифр мантиссы в десятичном представлении меньше суммы цифр мантисс в десятичном представлении множителей, вычитаем из порядка единицу

2.4 Проверка: если результирующий порядок меньше минимального, присваиваем результату ноль

2.5 Округляем число с возможным прибавлением единицы к порядку (П.3)

**3. Округление мантиссы**

3.1. Преобразуем мантиссу типа longint в строку десятичных цифр

3.2 Если длина строки превышает максимальное количество разрядов мантиссы, продолжаем округление

3.3 Преобразуем 30 первых символов строки в тип longint.

3.4 Если 31-й символ строки больше или равен символу «5», прибавляем к новой мантиссе единицу

3.5 Если количество символов в новой мантиссе больше максимального, присваиваем мантиссе единицу, устанавливаем флаг переполнения равным 1

3.6 Возвращаем значение флага переполнения

**4. Вывод результата**

4.1 Преобразуем мантиссу типа longint в строку десятичных цифр

4.2 Выводим строку десятичных цифр и порядок в указанном формате или ±inf в случае переполнения порядка сверху и +0.0E+0 в случае переполнения порядка снизу

**Ограничения**:

При вводе числа нельзя вводить пробельные символы кроме символа переноса строки сразу после самого числа.

**Функциональные тесты**:

**Позитивные**

|  |  |  |  |
| --- | --- | --- | --- |
| Первый множитель | Второй множитель | Результат | Описание |
| 1 | 1 | +0.1E+1 | умножение 2-х единиц |
| 10e10 | 3e0 | +0.3E+12 | с экспоненциальной частью без точки |
| 1.1E3 | 1.2E2 | +0.132E+6 | с экспоненциальной частью и точкой |
| -1.0 | -1e0 | +0.1E+1 | с точкой без экспоненциальной части |
| |  | | --- | |  | | -1.0 | |  | | 10e-1 | -0.1E+1 | отрицательное число, отрицательный порядок |
| -999.01 | 545.e3 | -0.54446045E+9 | ноль сразу после точки |
| 1 | 0 | +0.0E+0 | одно из чисел нулевое |
| -0.1e-99998 | 1e-1 | -0.1E-99999 | результат перемножения имеет минимальный порядок |
| 0.1e99998 | 10 | +0.1E+99999 | результат перемножения имеет максимальный порядок |
| 0.1e99998 | 100 | +inf | переполнение порядка сверху в результате (+inf) |
| -0.1e99998 | 100 | -inf | переполнение порядка сверху в результате (-inf) |
| -0.1e-100000 | 100 | +0.0E+0 | переполнение порядка снизу при вводе |
| -0.1e-99998 | 1e-2 | +0.0E+0 | переполнение порядка снизу в результате |
| -0 | 0e11 | +0.0E+0 | отрицательный ноль, нулевая мантисса |
| 1e100000 | 1e-10 | +inf | переполнение порядка сверху при вводе (+inf) |
| 1e100000 | -1e99999 | -inf | переполнение порядка сверху при вводе (+inf) |

**Негативные**

|  |  |  |
| --- | --- | --- |
| Первый множитель | Второй множитель | Описание |
| a | 12.3 | Первый множитель является буквой |
| -12.re10 | 78.2 | Буква в дробной части |
| 12.3e4e7 | 123 | Две буквы «e» |
| 123e-+12 | 45 | Два разных знака в порядке |
| .1e--1 | 321 | Два знака «-» в порядке, нет целой части |
| 1.3e--1 | 56 | Два знака «-» в порядке, есть целая часть, есть дробная |
| 1.1e++1 | 123 | Два знака «+» в порядке |

**Вывод:**

Если программисту необходим тип данных, выполняющий функции числа с плавающей точкой, с точностью, превышающей точность стандартных типов, он может использовать массив, каждый элемент которого будет являться разрядом соответствующего числа в десятичной системе счисления. Для уменьшения количества памяти, выделяемой под каждый объект нового типа, можно повысить эффективность использования памяти, создав функции для работы с массивом, каждый элемент которого будет являться разрядом числа в системе счисления с основанием 2^N, где N – размер элемента массива в битах.

**Контрольные вопросы:**

1. Каков возможный диапазон чисел, представляемых в ПК?

Максимальный диапазон чисел может быть ограничен лишь количеством памяти, выделенным под хранение этого числа. Если, например, тип беззнакового целого числа занимает в памяти 128 бит, то максимальное хранимое значение равно 2^128-1.

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

Точность представления чисел с плавающей точкой зависит от длины мантиссы и конкретного значения порядка. Большему значению порядка соответствует меньшая точность, так как максимальная длина мантиссы не изменяется.

3. Какие стандартные операции возможны над числами?

В зависимости от реализации структуры данных, возможны следующие операции:

Сложение, вычитание, умножение, деление, возведение в степень, взятие логарифма и т.д.

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

Программист может использовать массив, в котором каждый разряд числа будет храниться в виде отдельного элемента.

Программист может использовать массив, в котором каждый элемент будет являться разрядом искомого числа в системе счисления с основанием 2^N, где N – размер элемента массива в битах.

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

В случае с массивом, возможно последовательное использование побитовых операций над каждым элементом этого массива.