**Описание условия задачи**

Смоделировать операцию умножения целого числа длиной до 30 десятичных цифр 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. Описание исходных данных

На вход принимается строка, содержащая целое число в формате ±T, и строка, содержащая вещественное число в формате ±m.nE±K, где

T – последовательность цифр целого числа

m, n – последовательность цифр, содержащихся в мантиссе

E – экспонента – обозначает порядок

K – последовательность цифр порядка

Ограничения:

1. Суммарная длина мантиссы (m+n) – до 30 значащих цифр
2. Количество цифр в целом числе – до 30
3. Количество цифр в порядке – до 5
4. Вводимые строки должны содержать только корректно записанные числа – без лишних пробельных символов. При вводе некорректного числа работа программы будет завершена.
5. Возможно явное указание знака целого и/или вещественного числа первым символом. Если знак не указан явно, то считается, что число положительное. Аналогичное ограничение сделано относится к порядку.
6. Знак «+», «-» может быть указан 1 раз в начале мантиссы, и 1 раз перед порядком. Использование знака в другом месте будет являться некорректным вводом.
7. Для указания порядка может использоваться как строчная, так и прописная буква e. Порядок может быть опущен (если знак порядка - «e» - не был указан.
8. Если знак порядка указан – обязательно
9. Для обозначения дробной части числа используется точка. Она может отсутствовать в числе.
10. При наличии десятичной точки в числе возможны варианты представления, в которых число начинается или оканчивается точкой или же находится между цифр.
11. Порядок – целое число, при его записи не может быть указана «.».
12. Использование точки в других местах или дублирование будет являться некорректным вводом.
13. Числа могут быть введены с использованием незначащих нулей, однако, если с незначащими нулями количество цифр числа превышает допустимую, то это будет являться некорректным вводом. Знак должен указываться до незначащих нулей.

Описание результатов программы:

Результатом работы программы является вещественное число, записанное в формате ±0.m1Е±K1, где

m1 – последовательность цифр мантиссы, до 30 знаков,

K1 - последовательность цифр порядка, до 5 знаков,

E(e) – экспонента – обозначает порядок

Если при умножении длина мантиссы превысит 30 знаков, то будет произведено округление полученного числа по следующему правилу - если 31-й разряд больше или равен 5, то к 30-му разряду добавляется единица, если меньше 5, то 31-й разряд отбрасывается.

1. **Описание задачи, реализуемой программой**

Программа выполняет умножение длинного целого числа на длинное вещественное число.

1. **Способ обращения к программе**

Для запуска программы необходимо из консоли вызвать исполняемый файл ./app.exe

1. **Описание возможных аварийных ситуаций и ошибок пользователя**
2. Некорректный ввод целого числа

* 1234512345123451234512345123451 – превышено максимально допустимое число знаков
* -1123- - некорректное количество знаков «-»
* 1233444+4445 – знак числа указан в некорректном месте
* 12232323230. – при записи целого числа использован символ, отличный от «+», «-» и десятичных цифр
* 12ef34 – аналогично предыдущему
* 0000+4234 – знак числа указан после незначащих нулей
* Пустой ввод

1. Некорректный ввод вещественного числа

* Пустой ввод
* 123 – введено целое, а не вещественное число
* .e4 – отсутствуют цифры в мантиссе
* 12.3e+ - не указан порядок
* 1234567890e-100000 – превышено максимально допустимое число знаков порядка
* 123.345.5 – некорректное количество знаков «.»
* 12o5 – использование некорректного символа
* 12e+5e4 - некорректное количество знаков «e»
* 1234567890123456789012345678901e10 – превышено максимально допустимое число знаков мантиссы
* 123e5.3 – дробный порядок

**Описание внутренних структур данных**

В программе реализована структура для хранения длинного целого числа – содержащего до 30 значащих цифр.

**typedef** **struct** long\_int

{

**int** sign;

**int** digits[MAX\_LEN\_INT];

**int** count\_digit;

} long\_int;

Структура содержит следующие поля:

1. sign – является целочисленной переменной, хранящим знак числа

Если sign == -1, то число отрицательное, если sign == 1, то число положительное

1. digits – массив целых чисел, в котором хранятся цифры введенного числа, от старшего разряда к младшему
2. count\_digit – целочисленная переменная, которая отображает, сколько цифр в числе

В программе реализована структура для хранения длинного вещественного числа – содержащего до 30 значащих цифр в мантиссе и до 5 цифр в порядке.

**typedef** **struct** long\_float

{

**int** sign;

**int** digits[MAX\_NUM\_LEN];

**int** exp;

**int** count\_digit;

} long\_float;

Структура содержит следующие поля:

1. sign – является целочисленной переменной, хранящим знак числа

Если sign == -1, то число отрицательное, если sign == 1, то число положительное

1. digits – массив целых чисел, в котором поразрядно хранится мантисса, от старшего разряда к младшему
2. exp – целочисленная переменная, в которой хранится порядок числа
3. count\_digit – целочисленная переменная, которая отображает, сколько цифр в мантиссе

**Описание функций**

1. Функции ввода длинных чисел

* int input\_long(char \*str, size\_t len\_str, size\_t max\_len)

Осуществляет ввод строки и проверяет ее корректность

1. Функции обработки длинного целого числа

* int process\_long\_int(long\_int \*num);

Обрабатывает длинное целое число – последовательно вызывает и анализирует успешность выполнения функций, описанных ниже

* int validate\_int(char \*str);

Проверяет корректность введенного длинного числа из строки str;

* void save\_int(long\_int \*num, char \*str, size\_t len\_str);

Сохраняет длинное целое число из строки в структуру

* void print\_int(long\_int \*num);

Выводит на экран длинное целое число в формате [-]num;

* void delete\_zeros\_int(long\_int \*num\_int);

Функция удаляет незначащие нули в начале числа;

1. Функции обработки длинного вещественного числа

* int process\_long\_float(long\_float \*num);

Обрабатывает длинное целое число – последовательно вызывает и анализирует успешность выполнения функций, описанных ниже

* int validate\_float(char \*str, char \*buf);

Проверяет корректность введенного числа, корректное число записывает в buf

* void check\_sign\_correct(char \*buf, char \*str, char exp, char \*is\_correct, size\_t i);

Проверяет, что знак указан на нужной позиции

* void check\_point\_correct(char \*buf, char \*str, int \*is\_point, int is\_exp, char \*is\_correct);

Проверяет, что «.» указана на нужной позиции

* void check\_exp\_correct(char \*buf, char \*str, char point, int is\_digit, int \*is\_exp, char \*is\_correct, size\_t i);

Проверяет, что «e» указана на нужной позиции

* char is\_all\_param\_incorrect(size\_t len, char \*sign, char \*buf, char point, int is\_exp, int is\_point);

Проверяет корректность всех параметров, после окончания парсинга строки

* int save\_float(long\_float \*num, char \*buf);

Сохраняет длинное вещественное число из строки buf в структуру

* void delete\_zeros\_left(long\_float \*num\_float);

Удаляет нули из начала мантиссы, увеличивая экспоненту в num

* void delete\_zeros\_right(long\_float \*num);

Удаляет нули в правой части мантиссы, уменьшая количество цифр в num

* int check\_sign(char sym, size\_t \*index);

Проверяет, какой знак содержится в sym, и в зависимости от этого возвращает -1 или 1, и изменяет index на 1 или 0 соответственно

* void print\_float(long\_float \*num);

Выводит длинное вещественное число num на экран

1. Функции, выполняющие математические операции над длинными числами

* int multiplication(long\_float \*num\_float, long\_int \*num\_int, long\_float \*result);

Выполняет умножение длинного целого num\_int на длинное вещественное num\_float и сохраняет результат в result

* void num\_round(long\_float \*result)

Выполняет округление числа, если длинна мантиссы превышает 30 знаков

* void remove\_overflow(long\_float \*result, size\_t i)

Удаляет переполнение в разряде числа, которое могло появиться после умножения/округления

1. Функции, обрабатывающие ошибки

* void process\_error(int rc);

Выводит информацию об ошибки в зависимости от кода ошибки;

**Описание алгоритма**

Умножение длинного целого числа на длинное вещественное число происходит в несколько этапов:

1. После обработки и проверки корректности ввода числа, прошедшие валидацию, приводятся к нормализованному виду (для целого числа подразумевается удаление лишних незначащих нулей), после чего записываются в соответствующие структуры.
2. Умножение целого числа на вещественное происходит аналогично алгоритму умножения двух целых чисел «в столбик». После чего результат умножения двух целых чисел сохраняется в мантиссу числа-результата. Знаки чисел перемножаются. Порядок итогового числа равен суммарному порядку двух перемножаемых чисел (Целое число можно рассматривать как вещественное, приведенное к нормализованному виду, у которого порядок равен количеству разрядов в числе).
3. Полученное число приводится к нормализованному виду и выводится на экран.

**Набор тестов**

1. Негативные тесты для целого числа (вещественное число вводится корректно)

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Входные данные** | **Выходные данные** |
| Пустая строка | 10.5 | Введена пустая строка |
| Число содержит только знак «-» | -  1.2e3 | Введено некорректное целое число |
| Число содержит только знак «+» | +  1.2e3 | Введено некорректное целое число |
| Некорректный символ | .  1.2e3 | Введено некорректное целое число |
| Вещественное число | 123.  1.2e3 | Введено некорректное целое число |
| Число содержит более одного символа «-» | -1234-5  1.2e3 | Введено некорректное целое число |
| Число содержит более одного символа «+» | +12434+2  1.2e3 | Введено некорректное целое число |
| Число с другими некорректными символами | 123122e34  1.2e3 | Введено некорректное целое число |
| Количество разрядов числа превышает допустимое (30) | 1234512345123451234512345123456  1.2e3 | Введено слишком длинное число |

1. Негативные тесты для вещественного числа (целое число вводится корректно)

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Входные данные** | **Выходные данные** |
| Пустая строка | 12 | Введена пустая строка |
| Число содержит только знак «-» | 12  - | Введено некорректное вещественное число |
| Число содержит только знак «+» | 12  + | Введено некорректное вещественное число |
| Число содержит только знак «.» | 12  . | Введено некорректное вещественное число |
| Целое число | 12  12 | Введено некорректное вещественное число |
| Число с другими некорректными символами | 13  12o5 | Введено некорректное вещественное число |
| После экспоненты не указан порядок | 12  12e | Введено некорректное вещественное число |
| Число содержит более одного знака порядка | 12  12e+5e4 | Введено некорректное вещественное число |
| Число содержит более одного символа «.» | 12  123.345.5 | Введено некорректное вещественное число |
| Дробный порядок | 12  123e5.3 | Введено некорректное вещественное число |
| Пропущен знак порядка | 12  123e 5 | Введено некорректное вещественное число |
| Не указана мантисса | 12  .e4 | Введено некорректное вещественное число |
| Количество разрядов мантиссы превышает допустимое (30) | 12  1234567890123456789012345678901e10 | Превышена допустимая длина мантиссы |
| Значение порядка превышает допустимое максимальное (99999) | 12  1234567890e100000 | Превышено допустимое число разрядов порядка |
| Значение порядка превышает допустимое минимальное (-99999) | 12  1234567890e-100000 | Превышено допустимое число разрядов порядка |
| Начальные данные корректны, значение порядка после выполнения умножения превышает допустимое максимальное (99999) | 100  1e99999 | Превышено допустимое число разрядов порядка в процессе вычисления |
| Начальные данные корректны, значение порядка после выполнения умножения превышает допустимое минимальное (-99999) | 1  0.0000001e-99999 | Превышено допустимое число разрядов порядка в процессе вычисления |

1. Позитивные тесты

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Входные данные** | **Выходные данные** |
| Умножение положительного целого на положительное "целое" | 3  15. | +0.45e+2 |
| Округление в меньшую сторону | 6  999999999999999999999999999999. | +0.599999999999999999999999999999e+31 |
| Умножение на 0 | 0  1234.12e+4 | +0.e+0 |
| Округление в большую сторону | 3  999999999999999999999999999999e0 | +0.3e+31 |
| В результате умножения получаются лишние нули "справа" | 30  999999999999999999999999999999e0 | +0.3e+32 |
| Умножение 30-ти разрядного на 30-ти разрядное число | 111111111111111111111111111111  999999999999999999999999999999. | +0.111111111111111111111111111111e+60 |
| Умножение целого положительного на положительное вещественное без указания порядка | 12332432432  123421424.2442442 | +0.15220863751533482614078944e+19 |
| Умножение целого на вещественное не в нормализованной форме с указанием положительного порядка | 24  123.23e5 | +0.295752e+9 |
| Умножение целого на вещественное не в нормализованной форме с указанием отрицательного порядка | 24  123.23e-5 | +0.295752e-1 |
| Умножение целого отрицательного на отрицательное вещественное | -123  -2e1 | +0.246e+4 |
| Умножение целого положительного на отрицательное вещественное | 123  -0.1e2 | -0.123e+4 |
| Умножение целого отрицательного на положительное вещественное | -123  0.1e2 | -0.123e+4 |
| Умножение на 1 | 1  99999999999999999999999999e5000 | +0.99999999999999999999999999e+5026 |
| порядок указан с E | 123  123.24E5 | +0.1515852e+10 |
| Максимальное значение порядка | 1  1.e99998 | +0.1e+99999 |
| Минимальное значение порядка | 1  0.01e-99998 | +0.1e-99999 |
| Положительное число указано с + | 123  +123.24e+5 | 123  +123.24e+5 |

**Вывод**

При работе с длинными числами, т. е. с числами, которые не помещаются в стандартные типы данных, необходимо самостоятельно реализовать структуру для хранения и обработки таких чисел, а также самостоятельно реализовать операции над ними. Для хранения таких чисел удобно использовать структуру, в которой отдельное поле будет хранить знак числа; следующее поле - массив, в котором хранятся цифры числа; поля, содержащие количество цифр и порядок. Операции удобно выполнять способом, аналогичным математическим - “в столбик”, т. е. поразрядно работать с числами.

**Ответы на контрольные вопросы**

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон значений числа зависит от размера области памяти, выделяемой под хранение переменной, от наличия знака в числе и

от типа представления числа.

Если под хранение целого положительного числа выделено 16 разрядов, то его максимальное значение не может превышать 216-1=65 535, если выделено 32 разряда, то максимальное значение составит 232 -1=4 294 967 295. Для 64 разрядов максимально возможное значение числа равно 264 - 1=18 446 744 073 709 551 615.

На 64 битной машине возможные значения чисел вещественных чисел находятся в диапазоне от 3.6 E –4951 до 1.1 E +4932.

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

Длина мантиссы определяет точность представления числа, а длина порядка ограничивает диапазон допустимых значений. Максимально под представление мантиссы отводится 52 двоичных разряда, а под представление порядка – двоичных 11 разрядов.

1. Какие стандартные операции возможны над числами?

Математические - сложение, вычитание, умножение, деление; Побитовые сдвиги, и, или, не; Логические (над целыми) - конъюнкция, дизъюнкция, инверсия; Операции сравнения;

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

Для хранения таких чисел может быть использован массив, например массив символов или чисел – для ввода и вывода числа, числовой массив – для обработки. Однако удобно использовать структуру для хранения всего числа, например знак мантиссы, мантисса, знак порядка, порядок.

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

Для выполнения операций над числами, выходящими за рамки машинного представления, можно последовательно выполнять необходимую операцию над цифрами числа, например, аналогично выполнению математических операций «в столбик».