

Условие задачи:

Смоделировать операцию умножения целого числа длиной до 40 десятичных цифр на действительное число в форме $m.nE K$, где суммарная длина мантиссы ($m + n$) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $0.m1E K1$, где $m1$ - до 40 значащих цифр, а $K1$ - до 5 цифр.

Техническое задание:

На вход подаются целое и действительное число:

- 1) Формат целого числа: число написано без пробелов, перед числом может быть знак или он может отсутствовать (что означает положительность числа). Знак (при наличии) пишется вплотную к числу, без пробела. Перед числом и после числа могут стоять пробельные символы. Перед числом могут стоять незначащие нули. Всего количество цифр в числе не должно превышать 40, не учитывая знак и незначащие нули.

Ввод целого числа: производится в первую очередь, после предложения его ввести в командной строке .

- 2) Формат действительного числа: число написано в формате $m.nE K$, где суммарная длина мантиссы ($m + n$) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Знак, число, символ «Е» или «е», знак порядка и сам порядок написаны без пробелов между ними, представление числовой части в виде «.234» или «234.» является правильным. Порядок также может представляться без знака, в этом случае он положительный. Перед числом и после него могут присутствовать пробельные символы. При вводе числа могут присутствовать незначащие нули перед числом (после знака) и в конце числа (перед «Е» или «е», при его наличии), если эти нули после точки и всех значащих цифр. В подсчете количества значащих цифр в мантиссе не участвуют незначащие нули, описанные в предыдущем предложении.

Ввод действительного числа: производится во вторую очередь, после предложения его ввести в командной строке.

На выходе мы получаем число в формате 0.m1E K1, где m1 - до 40 значащих цифр, а K1 - до 5 цифр. В числе при выводе нет пробелов. В случае положительности числа, оно выводится без знака.

Задача, выполняемая программой:

Программа перемножает два «больших» числа и выводит результат в формате, удобном для понимания человеком.

Способ обращения к программе:

Чтобы обратиться к программе пользователю необходимо запустить файл test.exe из папки, в котором он находится.

Возможные аварийные ситуации или ошибки пользователя:

Ситуация	Вывод	Код возврата
Пустая строка	Ошибка! Была введена пустая строка.	3
Переполнение строки	Ошибка! Неправильная длина строки.	1
Ошибка в формате целого числа	Ошибка! Некоторые проблемы со вводом целого числа.	2
Ошибка в формате действительного числа	Ошибка! Формат действительного числа ошибочен.	4
Переполнение порядка итогового числа	Ошибка! Переполнение порядка, результат не может быть выведен.	6
Неверное количество точек в действительном числе	Ошибка! Некоторые проблемы с точками в действительном числе.	5

Внутренние структуры данных:

```
M_LEN = 40;  
M_ORD = 5;  
M_AMOUNT = 30;
```

В программе множество раз используются массивы символов для хранения строк с числом, например:

```
char buf[M_LEN + 100];
```

Хранит число при его вводе, в том виде, котором мы его ввели.

```
char no_nules[M_LEN + 10];
```

Хранит целое число без незначащих нулей и знака. Если перед или после числа есть пробельные символы они также не добавляются в этот массив.

```
char order[M_ORD + 10];
```

Хранит порядок действительного числа со знаком.

```
char mantis[M_AMOUNT + 10];
```

Хранит мантиссу действительного числа с незначащими нулями, если все значащие цифры после точки.

```
char no_nules_mantis[M_AMOUNT + 10];
```

Хранит мантиссу если в части со значащими цифрами присутствует точка.

```
char final_number[M_AMOUNT + 10];
```

Хранит вещественную часть действительного числа как строку целых цифр, образуется на основе одного из 2 предыдущих массивов, в зависимости от ситуации.

```
int first_rev_fours[10] = {0};
```

```
int second_rev_fours[8] = {0};
```

Эти два массива хранят значащие цифры первого и второго числа, в формате, удобном для перемножения четверками.

```
char mantis[M_AMOUNT + 21];
```

Этот массив содержит результат перемножения четверками, он не округлен. В дальнейшем в этом же массиве будет содержаться округленный результат.

Алгоритм:

1. Программа считывает целое число в строку.
2. Проверяется формат целого числа. Если формат верен, переходим к следующему шагу.
3. Целое число разбивается на четверки и располагается в целочисленном массиве в перевернутом виде (последняя четверка – первая, первая - последняя).
4. Программа считывает действительное число в строку.
5. Проверяется формат действительного числа. Если формат верен переходим к следующему шагу.
6. Переписываем значащие цифры числа в строку. Полученную строку разделяем на ранее упомянутые четверки.
7. Перемножаем два числа по четверкам.
8. Переписываем результат в строку, в четверки, где это необходимо, добавляем ведущие нули.
9. Округляем число в строке до необходимого количества символов, если символов больше чем 40.
10. Меняем порядок числа, если это необходимо.
11. Выводим число в зависимости от его знака.

Тесты:

В приведенных примерах знак «_» означает пробел.

Целое	Вещественное	Код возвр.	Что проверяется	Вывод
Пустая строка		3	Пустая строка	Ошибка! Была введена пустая строка.
45663	Пустая строка	3	Пустая строка	Ошибка! Была введена пустая строка.
-4[40 цифр]		1	Переполнение	Ошибка! Неправильная длина строки.
437467	0.0005[30 цифр]	1	Переполнение	Ошибка! Неправильная длина строки.
3657_37648		2	Формат целого числа	Ошибка! Некоторые проблемы со

				вводом целого числа.
536565883495	-_437584.3758	4	Формат действ-го числа	Ошибка! Формат действительного числа ошибочен.
32564[40 цифр]		1	Переполнение	Ошибка! Неправильная длина строки.
346756	37284[31 цифра]	1	Переполнение	Ошибка! Неправильная длина строки.
8657.42367		2	Формат целого числа	Ошибка! Некоторые проблемы со вводом целого числа.
+23647264	+3746.48263\374	4	Формат действ-го числа	Ошибка! Формат действительного числа ошибочен.
+ -364434		2	Формат целого числа	Ошибка! Некоторые проблемы со вводом целого числа.
2546642	--4637.47376	4	Формат действ-го числа	Ошибка! Формат действительного числа ошибочен.
+_34682		2	Формат целого числа	Ошибка! Некоторые проблемы со вводом целого числа.
45743	2456.2345E+4572 8563	1	Переполнение	Ошибка! Неправильная длина строки.
43765	0.00034637e++13	4	Формат действ-го числа	Ошибка! Формат действительного числа ошибочен.

4725	8.E999999	1	Переполнение	Ошибка! Неправильная длина строки.
+364791035756384 6586970035462817 494326734	0.0000476375e-99	0	Максимальное количество цифр в целом числе	0.1737773296584 47741786785064 3599683859898E -63
-349573	+47353758354657 9.36582946574859 5e+123	0	Максимальное количество цифр в действ. числе	-0.165535953693 12838865110383 0133599935E144
45	0.0001E+99999	0	Максимальное количество цифр в порядке	0.45E99997
-32743567	-000000.3463765	0	Незначащие нули в действительно м числе	0.1134160213497 55E8
____0003446____	___0.00034756457 e45___	0	Пробелы после и перед числом	0.119770750822 E46
48564876	0.35634554758646 597532666675767 4000000000	0	Незначащие нули после точки в действительно м числе	0.1730587733168 88193699586305 79759858424E8
2345667	0.01e99999	6	Переполнение порядка в результате умножения	Ошибка! Переполнение порядка, результат не может быть выведен.
1	0.000000004642e- 99999	6	Переполнение порядка в результате умножения	Ошибка! Переполнение порядка, результат не может быть выведен.
1	0.0000023e-99995	6	Переполнение порядка в результате	Ошибка! Переполнение порядка,

			умножения	результат не может быть выведен.
3654375	3265.653456.4645 6	5	Лишняя точка в действ. числе	Ошибка! Некоторые проблемы с точками в действительном числе.
2000	0.01e-99999	0	Нормализация только итогового результата	0.2000E-99997
9999999999999999 9999999999999999 99999999	2	0	Проверка округления	0.200000000000000 0000000000000000 000000000000000E 41

Вывод:

В результате работы была реализована программы для перемножения «больших» (выходящих за диапазон, предоставленный ЭВМ) чисел и вывода их в понятном пользователю формате.

Контрольные вопросы:

Каков возможный диапазон чисел, представляемых в ПК?

Все зависит от разрядности процессора ПК, если он 64-разрядный, то целые числа меняются от -2^{32} до 2^{32} , вещественные от -3.4×10^{38} до 3.4×10^{38} .

Какова возможная точность представления чисел, чем она определяется?

Точность числа определяется мантиссой, она может достигать 52 символа.

Какие стандартные операции возможны над числами?

- Сложение
- Вычитание
- Умножение
- Деление
- Сравнение

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

Можно выбрать массив символов (строку). Для обработки вещественного числа можно использовать структуру с полями: знак, мантисса, экспонента, порядок числа.

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

Чтобы провести операции над такими числами нужно использовать методы типа «в столбик». Когда мы берем меньшие части чисел совершаем с ней операцию и получаем часть ответа.