

## Работа № 1

### «Длинная» арифметика. Тип данных – массив

**Цель работы:** реализовать арифметические операции над числами, выходящими за разрядную сетку персонального компьютера, выбрать необходимые типы данных для хранения и обработки указанных чисел.

### Краткие теоретические сведения

Современный персональный компьютер (ПК) часто применяется для выполнения вычислений. При этом, для хранения исходных данных и результатов вычислений обычно используются стандартные числовые типы данных (целые и вещественные, в том числе, удвоенной точности). Каждый тип данных характеризуется определенным диапазоном значений чисел, который, в свою очередь, зависит от размера области памяти, выделяемой под хранение переменной этого типа, от наличия знака в числе и от типа представления числа (целое или вещественное).

*Целое число X* со знаком представляется в ПК следующим образом:

если  $X \geq 0$ , то число записывается как беззнаковое,

если  $X < 0$ , то число переводится в дополнительный код и записывается как  $2^k$

$-|X|$ , где  $k$  – количество разрядов, выделенное под представление числа.

Таким образом, если под хранение целого положительного числа выделено 16 разрядов, то его максимальное значение не может превышать  $2^{16}-1=65\,535$ , если выделено 32 разряда, то максимальное значение составит  $2^{32}-1=4\,294\,967\,295$ . Для 64 разрядов максимально возможное значение числа равно  $2^{64}-1=18\,446\,744\,073\,709\,551\,615$ .

Для 64-разрядного процессора принципиально невозможно использовать больше 20 десятичных разрядов для представления числа, поэтому при необходимости обрабатывать числа большей размерности (например, при выполнении астрономических расчетов) хранение данных и их обработку должен реализовать программист.

*Вещественные числа* обычно хранятся и используются в представлении с плавающей точкой в виде:

$$X = M * E^p,$$

где  $M$  – мантисса со знаком,  $E$  – основание (10 или 16),  $p$  – целый порядок со знаком.

Если десятичная точка расположена в мантиссе перед первой значащей цифрой числа, то при фиксированном количестве разрядов, отведённых под мантиссу, обеспечивается возможность сохранить максимальное количество значащих цифр, то есть обеспечить максимальную точность представления числа в ПК. Из сказанного следует, что мантисса должна быть *правильной дробью*, первая цифра которой отлична от нуля, т.е.  $M$  находится в интервале  $[0.1, 0.99...9]$ . Такое представление вещественных чисел называется *нормализованным*.

Таким образом, длина мантиссы определяет точность представления числа, а длина порядка ограничивает диапазон допустимых значений. При этом, если мантисса выходит за разрядную сетку ПК, то происходит ее округление. При выходе за заданный диапазон величины порядка могут возникнуть проблемы, связанные с переполнением порядка (при положительном порядке) или получением машинного нуля (при отрицательном порядке).

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

В том случае, если требуется очень высокая точность вычислений (не ниже 20–30 знаков после десятичной точки) или необходимо обрабатывать числа с большим порядком (превышает 5000) (в навигационных системах или в системах наведения), то задача выбора необходимых структур для хранения и обработки данных и реализации необходимых операций над ними также возлагается на программиста.

### Задание

Составить программу умножения или деления двух чисел, порядок которых находится в диапазоне от  $-99999$  до  $+99999$  (т.е. имеет не более 5 разрядов), а длина мантиссы не превышает, например 30 разрядов. (Длина мантиссы дана в варианте задания и могут отличаться от данного примера)

Программа должна осуществлять ввод чисел в указанном диапазоне значений и выдавать результат в нормализованной форме  $\pm 0.m1 \text{ E } \pm K1$ , где число  $m1$  определено до 30 значащих цифр, число  $K1$  – до 5 цифр. При невозможности произвести вычисления должно выдаваться соответствующее сообщение.

При получении результата с порядком, большим по модулю, чем указано, необходимо сообщить о достижении машинного нуля или машинной бесконечности (переполнение порядка)

### Указания к выполнению работы

При выполнении лабораторной работы следует обратить внимание, что при хранении чисел в оперативной памяти компьютера необходимо обеспечить следующий формат их представления (рис. 1,2):

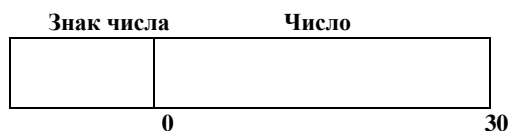


Рис.1. Представление целого числа



Рис.2. Представление вещественного числа

При вводе целого числа не допускается ввод точки и порядка. Возможен ввод знака перед числом. В случае положительного числа знак опционален.

Ввод вещественного числа НЕ обязательно осуществлять в нормализованном виде. При этом количество значащих цифр не должно превышать указанное в задании.

Ведущие нули (не значащие – до точки и/или до первой цифры) в расчете длины числа не учитываются. Значащие нули после точки учитываются при подсчете длины числа.

Десятичное число может представляться без точки: 123 (как целое).

При наличии точки в числе возможны следующие варианты его представления:

.00025, +123001., – 123.456.

Также допускается представление числа в экспоненциальной форме:

1234567 E –20, 1234567 E 20 или 123.4567 E23.

Так как работа ведется в консольном режиме и с длинными числами, хорошо бы предусмотреть «линейку» над вводимым числом, например:

>+-----1-----2-----3-----4-----5

**В программе должна быть реализована возможность ввода чисел в любом из перечисленных представлений.**

Результат при выдаче на печать должен быть нормализован в виде: знак 0.мантисса Е знак порядок. Первая цифра мантиссы не является нулем.

Указанный формат не имеет стандартного представления в машине, поэтому программист сам должен выбрать типы данных, используя которые возможно осуществить необходимые действия (ввод, вывод, обработку) с данными числами. Наиболее предпочтительным типом для этого является массив, например, массив символов или чисел – для ввода и вывода числа, числовой массив – для обработки. Но, можно использовать структуру для хранения всего числа, например, знак мантиссы, мантисса, знак порядка, порядок. Можно разбить мантиссу на несколько частей, обрабатывая их, а затем «склеивая» число. Выбор – за программистом.

Проще производить вычисления, если перед обработкой числа нормализованы, т.е. приведены к такому виду, когда после нуля следует значащая цифра. **При нормализации порядок входных данных может выйти за пределы указанного диапазона, но при правильных исходных данных, это не должно приводить к ошибке.** Для этого необходимо предусмотреть дополнительные разряды в тех структурах данных, которые предназначены для хранения и обработки промежуточных результатов вычисления.

Если в результате умножения или деления чисел длина мантиссы стала больше указанного количества 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). При этом может возникнуть циклический поразрядный перенос из младшего разряда в старший с коррекцией порядка. Например, для 5-ти разрядов:

$$99999 \text{ E } 01 + 00008 \text{ E } 01 = 100007 \text{ E } 01 \rightarrow 10001 \text{ E } 02 .$$

Так же как нет стандартных типов для хранения таких больших чисел, так нет и стандартных арифметических операций для их обработки, поэтому необходимо разработать эти операции самостоятельно. Если мантисса хранится в массиве, то, удобно использовать алгоритм умножения и деления в «столбик».

## Примеры обработки и представления длинных чисел в массиве цифр

Рассмотрим наиболее простое для понимания представление длинных чисел в виде массива значащих цифр (каждая цифра длинного числа является элементом массива чисел)

### Пример 1. Умножение длинного числа (3123456789) на короткое (7)

Таблица 1. Умножения длинного числа на короткое в «столбик»

№	Значение разрядов в массиве									
1	3	1	2	3	4	5	6	7	8	9
2	21	7	14	21	28	35	42	49	56	63
3	21	8	14+2=16 <-1	21+3=24 <-2	28+3=31 <-3	35+4=39 <-3	42+5=47 <-4	49+6=55 <-5	56+6=62 <-6	3 <-6
4	21	8	6	4	1	9	7	5	2	3

Первая строка таблицы 1 – содержимое массива, представляющего цифры исходного числа.

На первом шаге каждая цифра длинного числа умножается на короткое число. Полученный массив представлен в строке 2 таблицы 1.

Далее в результирующий массив (строка 4 таблицы 1) записывается последняя цифра числа из массива произведений цифр, а старший разряд произведения суммируется с значением из предыдущей ячейки (старший разряд) (показано в строке 3 таблицы 1).

В примере обработка числа идет справа налево, но для удобства работы с массивом, можно хранить цифры числа в «перевернутом» виде, начиная с младших разрядов. Об этом необходимо помнить при выводе числа.

### Пример 2. Умножение «длинного» числа на «длинное»

Числа хранятся в массивах цифр, но в «перевернутом» виде, где старшие разряды хранятся слева, а младшие – справа. Например, числа: 456789 \* 62345 хранятся в виде: 987654\*54326.

При умножении двух длинных чисел в «столбик», первое число умножается на каждую цифру второго. Результат умножения на  $i$ -ю цифру прибавляется к общему результату (в массиве чисел) со сдвигом на  $i+1$ .

Таблица 2. Умножение длинного числа на длинное в «столбик»

	Значение разрядов в массиве									
<b>5</b>	45	40	35	30	25	20				
<b>4</b>		36	32	28	24	20	16			
<b>3</b>			27	24	21	18	15	12		
<b>2</b>				18	16	14	12	10	8	
<b>6</b>					54	48	42	36	30	24
	<b>45</b>	<b>76</b>	<b>94</b>	<b>100</b>	<b>140</b>	<b>120</b>	<b>85</b>	<b>58</b>	<b>38</b>	<b>24</b>
	4 ->	80	102	110	151	135	98	67	44	28
		8->	10->	11->	15->	13->	9->	6->	4->	
	<b>5</b>	<b>0</b>	<b>2</b>	<b>0</b>	<b>1</b>	<b>5</b>	<b>8</b>	<b>7</b>	<b>4</b>	<b>28</b>

Результат = 28'478'510'205

### Пример 3. Деление чисел в столбик

Таблица 3. Алгоритм деления двух чисел в столбик

<b>1</b>	<b>1234567</b>   <u>3748</u> _____	Определить неполное делимое. Сравнить старшие разряды у делителя и делимого. Если старший разряд делимого больше, то длина неполного делимого равна длине делителя, если меньше, то длина неполного делителя на 1 больше, если равны, то сравнить следующие разряды
<b>2</b>	<b>12345, 3748 (: 1000)</b> <b>12 : 3 = 4</b>	разделить неполное делимое и делитель на $10^{(k-1)}$ , где $k$ – длина делителя. Получить их частное $t$
<b>3</b>	<b>4 * 3748 = 14992</b> <b>12345 &lt; 14992:</b> <b>4 - 1 = 3</b> <b>3 * 3748 = 11244</b> <b>12345 – 11244 =</b> <b>1101</b>	a. Умножить $t$ на делитель. b. Если результат больше неполного делимого, то уменьшить $t = t - 1$ , иначе $c$ c. умножить $t$ на делитель d. Найти разность делимого и делителя
<b>4</b>	<b>11016</b>	Сносим следующую цифру делимого. Если ее нет, ставим 0 и изменяем порядок
Далее повторить действия с п.2. до тех пор, пока неполное делимое и остаток не будут равны 0 или пока количество значащих цифр не будет равным длине результата + 1. Затем, при необходимости,		

округлить результат.
----------------------

### Обработка длинных чисел при хранении нескольких цифр числа в каждом элементе массива

Представление чисел можно сделать более эффективным, если в каждый элемент массива (типа int) вводить числа до  $10^4$ .

Например, число 123456789 можно представить:

$$1\ 2345\ 6789 = 0001 * 10^8 + 2345 * 10^4 + 6789 * 10^0 = \\ = 1 * (10^4)^2 + 2345 * (10^4)^1 + 6789 * (10^4)^0$$

Таким образом, представляем наше число в системе счисления с основанием 10000. Для удобства вычисления можно представить число в виде массива, записывая элементы в обратном порядке, где индекс элемента массива соответствует степени при преобразовании: | 6789 | 2345 | 0001 |

Выбор системы счисления опционален. В данном примере для хранения длинного числа предусматривался массив чисел типа int. Так что при умножении двух четырехзначных чисел не произойдет выход за границы типа.

### Пример 4. Сложение двух чисел.

$$A = 2345'6781'3245 \quad B = 348'7654'8749'1384$$

Таблица 4. Пример работы алгоритма сложения двух длинных чисел, хранящихся в массиве по тетрадам

A	3245	6781	2345	0
B	1384	8749	7654	348
C	4629	15530	9999	348
		1 ->	1 ->	
C	4629	5530	0000	349

$$C = 349'0000'5530'4629$$

Числа по тетрадам записываются в массивы в обратном порядке, слева младшие разряды, справа – старшие.

Складываются соответствующие элементы массивов. В случае, когда длина суммы становится больше 4, в текущем разряде остается только 4 младшие цифры. Пятая, старшая цифра, суммируется с числом следующего разряда.

### Пример 5. Умножение «длинного» числа на «короткое»

$$A = 2345'6781'3245 \quad B = 756$$

Таблица 5. Умножение длинного числа на короткое. Хранение по тетрадам

A	3245	6781	2345	0
B	756			
C	3245 * 756 = 245'3220	6781 * 756 = 512'6436 +245 = 512 6681	2345 * 756 = 177'2820 + 512 = 177 3332	
	245 ->	512 ->	177 ->	
C	3220	6681	3332	177

$$C = 177\ 3332\ 6681\ 3220$$

Алгоритм аналогичен представленному в примере 1. Здесь число представлено по тетрадам в СС<sub>10000</sub>. При этом число хранится в перевернутом виде, то есть младшие разряды слева, старшие справа.

Каждая цифра числа (в СС<sub>10000</sub>) умножается на «короткое число», размером не более 1 тетрады. Далее, в случае, когда результат умножения больше 9999, младшие разряды сохраняются в результирующем массиве, а старшие разряды суммируются со следующим элементом массива.

### Пример 6. Умножение «длинного» числа на «длинное»

$$A = 1'2345'6789'1234 \quad B = 12'3456$$

Таблица 6. Умножение длинного числа на длинное в «столбик»

	<b>1234</b>	<b>6789</b>	<b>2345</b>	<b>1</b>	
<b>3456</b>	1234 * 3456 426'4704	6789 * 3456 2346'2784	2345 * 3456 810'4320	1 * 3456 3456	
<b>12</b>		1234 * 12 1'4808	6789 * 12 8'1468	2345 * 12 2'8140	1 * 12 12
	426 4704	2347 7592 + 426	818 5788 + 2347	3 1596 + 818	12 + 4
	426 ->	2347 8018 2347 ->	818 8135 818 ->	3 2414 3->	15
	4704	8018	8135	2414	15

$$C = 15\ 2414\ 8135\ 8018\ 4704$$

Алгоритм работы аналогичен алгоритму, описанному в примере 2.



Числа представлены тетрадами и записываются в массив в обратном порядке, слева младшие разряды, справа - старшие.

Каждый следующий разряд второго множителя сдвигается на  $i+1$  в результирующем массиве. На каждой позиции результирующего массива остаются только 4 младших разряда, а все старшие разряды суммируются со следующим элементом массива.

Все логически завершенные фрагменты алгоритма (ввод, вывод, обработка и т.п.) необходимо оформить в виде подпрограмм. Проект должен быть многофайловым.

При разработке интерфейса программы следует предусмотреть:

- указание операции, производимой программой,
- указание формата и диапазона вводимых данных,
- указание формата выводимых данных,
- наличие пояснений при выводе результата.

При тестировании программы необходимо:

- проверить правильность ввода и вывода данных (т.е. их соответствие требуемому формату);
- обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»);
- проверить правильность выполнения операций;
- реализовать округление при превышении разрядности мантиссы;
- отследить возникновение переполнения и/или машинного нуля.

Необходимо также протестировать программу на границах допустимых значений данных, задавая самое большое и самое маленькое число в заданном диапазоне представления.

## Содержание отчета

Отчет по лабораторной работе должны содержать:

- 1) описание условия задачи;
- 2) описание ТЗ, включающем **внешнюю спецификацию**:
  - a. описание исходных данных (не языка программирования, а данных, понятных пользователю) и результатов (**то есть, типы, форматы, точность, способ передачи, ограничения**);
  - b. описание задачи, реализуемой программой;
  - c. способ обращения к программе;
  - d. описание возможных аварийных ситуаций и ошибок пользователя.
- 3) описание внутренних СД (на языке программирования);
- 4) описание алгоритма (в любом виде);
- 5) набор тестов, с **указанием, что проверяется**;
- 6) выводы по проделанной работе

Кроме того, в отчете должны быть даны ответы на следующие вопросы.

1. Каков возможный диапазон чисел, представляемых в ПК?
2. Какова возможная точность представления чисел, чем она определяется?
3. Какие стандартные операции возможны над числами?
4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?
5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Отчет представляется в электронном или печатном виде.