= 8 + 2 + 4 =
10 + 4 = 14 .
As students grow older , they commit more facts to memory , and learn to derive other facts rapidly and fluently . Many students never commit all the facts to memory , but can still find any basic fact quickly .

= = = = Carry = = = =

The standard algorithm for adding multidigit numbers is to align the addends vertically and add the columns , starting from the ones column on the right . If a column exceeds ten , the extra digit is " carried " into the next column . For example , in the addition 27 + 59
1
27
+ 59
? ? ? ?
86
7 + 9 = 16 , and the digit 1 is the carry . An alternate strategy starts adding from the most significant digit on the left ; this route makes carrying a little clumsier , but it is faster at getting a rough estimate of the sum . There are many alternative methods .

= = = = Addition of decimal fractions = = = =

Decimal fractions can be added by a simple modification of the above process . One aligns two decimal fractions above each other , with the decimal point in the same location . If necessary , one can add trailing zeros to a shorter decimal to make it the same length as the longer decimal . Finally , one performs the same addition process as above , except the decimal point is placed in the answer , exactly where it was placed in the summands .
As an example , 45 @.@ 1 + 4 @.@ 34 can be solved as follows :
4 5 . 1 0
+ 0 4 . 3 4
? ? ? ? ? ? ? ? ? ? ? ?
4 9 . 4 4

= = = = Scientific notation = = = =

In scientific notation , numbers are written in the form <formula> , where <formula> is the significand and <formula> is the exponential part . Addition requires two numbers in scientific notation to be represented using the same exponential part , so that the significand can be simply added or subtracted .
For example :
<formula>

= = = Addition in other bases = = =

Addition in other bases is very similar to decimal addition . As an example , one can consider addition in binary . Adding two single @-@ digit binary numbers is relatively simple , using a form of carrying :
0 + 0 ? 0
0 + 1 ? 1
1 + 0 ? 1