Functions with Multiple Arguments

The earlier discussion of functions centered around functions of the form f(x)=y, that is, functions where we had one input value, x, and one output value y. The value of x came from the domain of the function and the value y is part of the range of the function. This is a convenient example of simple functions. However, we have actually seen more complex functions in our earlier presentations. Two such more complex functions are gcf(x_1,x_2) and lcm(x_1,x_2). In these cases the function requires two arguments, and for the moment we will restrict the domain of these functions to positive integers, that is the values 1, 2, 3, 4, and so on. As we have seen, gcf(12,18) is 6 and lcm(12,18) is 36. We see that the idea of a function has not changed; a value or now values go into the function and the function produces an output value. Let us look at these and other functions that accept more than one argument.

gcf(x_1,x_2) is a function that produces the greatest common factor (often called the greatest common divisor) of two numbers. We have seen examples of this function before and we even worked out multiple methods for calculating gcf(x_1,x_2), including the Euclidean algorithm. In those discussions we did not include using negative values for one or both arguments. The gcf(x_1,x_2) function is available in many computer languages, and in all cases it will behave as expected if the arguments are positive integers. Different implementations of the gcf(x_1,x_2) treat negative integer arguments differently. In some cases the value is undefined, in some cases the function operates as if both arguments were positive, and in some cases the function has defined a meaning for the gcf(x_1,x_2) that produces a negative value in some cases.

Naturally, care must be taken to understand how the implementation of gcf(x_1,x_2) is handled in the language you are using.

lcm(x_1,x_2) is a function that produces the least common multiple of two integer values. We have seen examples of this function before and we even worked out multiple methods for calculating lcm(x_1,x_2), including deriving it using the gcf(x_1,x_2). The lcm(x_1,x_2) function is available in many computer languages, and in all cases it will behave as expected if the arguments are positive integers. Different implementations of the lcm(x_1,x_2) treat negative integer arguments differently. In some cases the value is undefined, in some cases the function operates as if both arguments were positive, and in some cases the function has defined a meaning for the lcm(x_1,x_2) that produces a negative value in some cases. Naturally, care must be taken to understand how the implementation of lcm(x_1,x_2) is handled in the language you are using.

min(x_1,x_2)

min(x_1,x_2,x_3,x_4,x_5,…,x_n) is a function that returns the lowest value represented by the arguments. In some computer languages the min function only exists in the form min(x_1,x_2), that is with two arguments. In other languages, however, the function has been expanded to accept as many arguments as you wish. Thus, in languages that have defined the min function, the expression min(4.2,16) returns the value 1.6, whereas, in languages with the expanded syntax, we could write min(4.2, 1.6,7.8,5.2,-1.6,6.9,-4.3,-0.4,2.7) and that instance of the function would return the value -4.3.

$\max(x_1,x_2)$

$\max(x_1,x_2,x_3,x_4,x_5,\ldots,x_n)$ is a function that returns the highest value represented by the arguments. In some computer languages the max function only exists in the form $\max(x_1,x_2)$, that is with two arguments. In other languages, however, the function has been expanded to accept as many arguments as you wish. Thus, in languages that have defined the max function, the expression max(4.2,16) returns the value 4.2, whereas, in languages with the expanded syntax, we could write max(4.2, 1.6,7.8,5.2,-1.6,6.9,-4.3,-0.4,2.7) and that instance of the function would return the value 7.8.