CS50 Libraries

Overview

Libraries are shared collections of code that programmers can use to work with one another. Libraries usually include functions that may be commonly used among programmers. For example, the C library string.h includes many useful premade functions to manipulate strings (see below). By allowing us to use functions that others have already written, libraries enable us to build off of the work of others and use their functions in our own programs, instead of reinventing those functions ourselves.

Key Terms

- library
- header file

Using Libraries

To use a functions from a library in C, remember to `#include` the **header file** (such as with `#include <math.h>`), which defines the library's functions, at the top of your source code file. When compiling your code, you'll also need to link the library so that the resulting object code knows how to execute the functions.

Some Common C Library Functions

In `ctype.h`:

`isalnum()` takes a `char` as input, and returns `true` if the character is alphanumeric and `false` otherwise
`isalpha()` takes a `char` as input, and returns `true` if the character is alphabetic and `false` otherwise
`islower()` takes a `char` as input, and returns `true` if the character is lowercase and `false` otherwise
`isupper()` takes a `char` as input, and returns `true` if the character is uppercase and `false` otherwise
`tolower()` takes a `char` as input, and returns the character converted to lowercase if possible. If it's not possible, it returns the original character unchanged.
`toupper()` takes a `char` as input, and returns the character converted to uppercase if possible. If it's not possible, it returns the original character unchanged.

In `math.h`:

`ceil()` takes a `double` as input, and returns the smallest integer that is not less than the input, as a `double`
`cos()`, `sin()`, and `tan()` each take a `double` as input, and return the cosine, sine, or tangent of the input
`floor()` takes a `double` as input, and returns the largest integer that is not greater than the input, as a `double`
`pow()` takes two `double`s as input, and returns the value of the first input raised to the second value power
`lround()` takes a `double` as input, and returns a `long int` representing a rounded version of the number
`log()`, `log10()`, and `log2()` take a `double` as input, and return the logarithm of the number (base e, base 10, and base 2, respectively)

In `stdio.h`:

`printf()` takes a `string` as input, and prints it to standard output, displaying it on the screen

In `stdlib.h`:

`atoi()` takes a `string` as input, and converts the `string` to an `int` if possible, returning the `int`
`rand()` returns a pseudorandom integer, and will usually be seeded with `srand()` first

In `string.h`:

`strlen()` takes a `string` as input, and returns the length of the string, not including the null terminator
`strcmp()` takes two `string`s as input, and returns `0` if they are equal, less than `0` if the first `string` comes before the second, and greater than `0` if the first string comes after the second one
`strstr()` takes two `string`s as input, and finds the first occurrence of the second `string` in the first.

There are many more functions defined in these libraries and other libraries, and it is often a good idea to explore the existing C libraries to see what functions are available to you so that you don't re-create code that you could use library functions for instead. Check out reference.cs50.net for more information on C Library Functions.