1. Passing a parameter by value allows the value to be manipulated, while the original variable is not affected. For example, using passing parameter by value, I can implement a helper function that takes in a value, do some calculations on the value and store the result under the same variable name, then return this value to the main function with the original value unchanged so I can use it again later.

2. Passing a parameter by reference allows the function to modify variables outside of it. This can be helpful when I need a function that, for example, converts the value according to a formula. By passing a reference to the function, I only need to call the function and do not need the function to return a new value.

3. I would expect pass-by-value to run faster when the size of the value is very small (smaller than the size of its memory address), then copying its value gets faster than finding its address and passing the address. It should also use less memory since the length of value is shorter than the length of memory address. For example, if memory address is 64-bit, copying and passing a 16-bit or 32-bit value should be faster and use less memory than passing a 64-bit memory address.

4. Pass-by-reference should, on the other hand, runs faster and use less memory with longer, larger values than pass-by-value. For example, copying a long string will take more time as opposed to finding its address and passing that into the function.