1. Go through the iterations of strcpy() given as an example in Kernighan & Ritchie, The C Programming Language, 2nd Edition, Prentice Hall PTR, 1988, p.88 - 106 and explain their philosophy of brevity vs. clarity.

Clarity in coding terms means how clear your code is. Whether it is one line or multiple lines, to clarify your code can seem silly at first glance but on returning to your piece of code whether it’s a few days after or months the silliness quickly goes away when you understand exactly what the code does. If you work hard on your code and are clear then you’ll rarely need to write comments on your code.

Brevity in code is how short your code is, it is designing your code to be concise and straight to the point, it does not necessarily mean that the code is better than code that has clarity, but brief code can sometimes have less errors than a longer alternative.

Example of Brevity in code..

cout << \*iter++ << endl;

instead of..

cout << \*iter << endl;

iter++;

2. Look at the code for strcpy() given on Moodle,, and explain the speed differences of the function implementations (why do some run faster than others). Note the C code given in this program is old and buggy, it needs to be updated to run in Visual Studio 2013 (for example there is no need any more to enum a bool type, and you will need to add

#ifdef \_MSC\_VER

#define \_CRT\_SECURE\_NO\_WARNINGS

#endif

at the start of your program – p.s what does this #define do?) Also you need to change the format specifier and cast the c\_time to a float for the /1000 to work in the printf’s.

The speed differentiates between the methods because each method does a different thing than each other. Some methods take longer to compile as they have more code inside them therefore taking longer than others or being quicker than others.

#define is a useful C component that allows the programmer to give a name to a constant value before the program is compiled