Recursive Function: Recursion can be described as a function calling itself with the goal of solving a problem. Typically recursion is much more suited for problems that may be structured repetitively, the issue is typically divided into small problems to be easier to solve. The recursive process is repeatedly called until the base case or condition is reached. This method would be considered slower than looping in a program, the overhead of calling functions would make it slower than iteration. The primary disadvantage of a recursive function in comparison to a normal loop would be the potential memory or performance issues.

Normal Loop: A loop on the other hand typically repeats the same task in a designated block until a certain condition has been reached ending the loop. For example you could make a loop program that counts until you reach the number 50. The condition in this case would be \* if (count >= 50) \*. This would be considered faster than a recursive function due to no overhead of calling functions. On the other hand normal loops can be much more cumbersome due to the increased amount of lines and potential issues maintaining them on larger scale program. Loops are also performed on the instruction level in contrast to recursive functions that are performed on the function level.