Problem 1

`Substring matching` is the process of determining whether shorter string (the substring) is contained within a longer string. Substring matching plays important roles in the reconstruction of an unknown DNA string from pieces, and in searching for interesting substrings within a known DNA string.

Python provides a `find (substring, start, end)` string method that returns the lowest index (integer) where the substring is found in the index range `start <= index < end`. The `start` and `end` arguments are optional, but for this exercise we will make them required (you will learn later how to handle optional arguments). If the substring is not found, -1 is returned.

Without using the `find` string method, write a function name `multi_find(some_string, sub_string, start, end)` that, instead of returning one integer index, returns a string that contains zero or more indices separated by commas. In this case, the string will contain digits representing the integer indices. If the substring is not found, an empty string is returned.