Complete the examples calls and then write a docstring description for each function. (If you get stuck, read the flip-side of this page for hints.)

(a) def function_a(s1: str, s2: str) -> bool:
   ```
   >>> function_a( , )
   True
   >>> function_a( , )
   False
   ```

   return s1 in s2 and s2 in s1

(b) def function_b(s1: str, s2: str) -> bool:
   ```
   >>> function_b( , )
   True
   >>> function_b( , )
   False
   ```

   return s1[0] == s2[0] and s1[-1] == s2[-1]

(c) def function_c(s1: str, s2: str) -> bool:
   ```
   >>> function_c( , )
   True
   >>> function_c( , )
   False
   ```

   return min(len(s1), len(s2)) == len(s1)

(d) def function_d(s1: str, s2: str) -> bool:
   ```
   >>> function_d( , )
   True
   >>> function_d( , )
   False
   ```

   return s1[0] == s1[-1] or s2[0] == s2[-1]

(e) def function_e(s1: str, s2: str) -> bool:
   ```
   >>> function_e( , )
   True
   >>> function_e( , )
   False
   ```

   return str(len(s1)) in s2
(i) Match each docstring description to the function it describes.

1. function_ : Return True iff s1 and s2 have the same first character and s1 and s2 have the same last character.
2. function_ : Return True iff the number representing the length of s1 occurs in s2.
3. function_ : Return True iff the first and last characters in s1 are the same, the first and last characters in s2 are the same, or both.
4. function_ : Return True iff s1 and s2 are the same string.
5. function_ : Return True iff s1 has a length that is less than or equal to s2’s length.

(ii) Which function(s) also need a precondition? Add the precondition(s).