**1.Given a string, return the string made of its first two chars, so the String "Hello" yields "He". If the string is shorter than length 2, return whatever there is, so "X" yields "X", and the empty string "" yields the empty string "".**

**first\_two('Hello') → 'He'**

**first\_two('abcdefg') → 'ab'**

**first\_two('ab') → 'ab'**

def first\_two\_chars(input\_string):

result\_string = input\_string[:2]

return result\_string

print(first\_two\_chars("Hell0"))

print(first\_two\_chars("abcdef"))

print(first\_two\_chars("sri"))

print(first\_two\_chars(" "))

**2.Given a string, return a new string made of 3 copies of the last 2 chars of the original string. The string length will be at least 2.**

**extra\_end('Hello') → 'lololo'**

**extra\_end('ab') → 'ababab'**

**extra\_end('Hi') → 'HiHiHi'**

def repeat\_last\_two\_chars(input\_string):

last\_two\_chars = input\_string[-2:]

result\_string = last\_two\_chars \* 3

return result\_string

input\_str = "Hello"

result = repeat\_last\_two\_chars(input\_str)

print(result)

**3. Given a string of even length, return the first half. So the string "WooHoo" yields "Woo".**

**first\_half('WooHoo') → 'Woo'**

**first\_half('HelloThere') → 'Hello'**

**first\_half('abcdef') → 'abc'**

def first\_half(input\_string):

half\_length = len(input\_string) // 2

result\_string = input\_string[:half\_length]

return result\_string

input\_str = "WooHooho"

result = first\_half(input\_str)

print(result)

**4. Given a string, return a version without the first and last char, so "Hello" yields "ell". The string length will be at least 2.**

**without\_end('Hello') → 'ell'**

**without\_end('java') → 'av'**

**without\_end('coding') → 'odin'**

def remove\_first\_and\_last\_char(input\_string):

result\_string = input\_string[1:-1]

return result\_string

input\_str = "Hello"

result = remove\_first\_and\_last\_char(input\_str)

print(result)

**5. Given two strings, a and b, return the result of putting them together in the order abba, e.g. "Hi" and "Bye" returns "HiByeByeHi".**

**make\_abba('Hi', 'Bye') → 'HiByeByeHi'**

**make\_abba('Yo', 'Alice') → 'YoAliceAliceYo'**

**make\_abba('What', 'Up') → 'WhatUpUpWhat'**

def combine\_strings\_ab(a, b):

result\_string = a + b + b + a

return result\_string

a = "sai"

b = "dinkar"

result = combine\_strings\_ab(a,b)

print(result)

**6. Given 2 strings, a and b, return a string of the form short+long+short, with the shorter string on the outside and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0).**

**combo\_string('Hello', 'hi') → 'hiHellohi'**

**combo\_string('hi', 'Hello') → 'hiHellohi'**

**combo\_string('aaa', 'b') → 'baaab'**

def combine\_strings\_short\_long\_short(a, b):

if len(a) < len(b):

result\_string = a + b + a

else:

result\_string = b + a + b

return result\_string

a = "Hi"

b = "Hello"

result = combine\_strings\_short\_long\_short(a,b)

print(result)

**7. Given 2 strings, return their concatenation, except omit the first char of each. The strings will be at least length 1.**

**non\_start('Hello', 'There') → 'ellohere'**

**non\_start('java', 'code') → 'avaode'**

**non\_start('shotl', 'java') → 'hotlava'**

def concat\_without\_first\_char(a, b):

result\_string = a[1:] + b[1:]

return result\_string

string\_a = "Hello"

string\_b = "World"

result = concat\_without\_first\_char(string\_a, string\_b)

print(result)

**8. Given a string, return a "rotated left 2" version where the first 2 chars are moved to the end. The string length will be at least 2.**

**left2('Hello') → 'lloHe'**

**left2('java') → 'vaja'**

**left2('Hi') → 'Hi'**

def rotate\_left2(input\_string):

rotated\_string = input\_string[2:] + input\_string[:2]

return rotated\_string

input\_str = "Hello"

result = rotate\_left2(input\_str)

print(result)