Subgoal Labelling and Strings

Write your names and roles here:

Example Mai has some code that returns the concatenation of two strings, but omits the first character of each. Her code looks like this:

```
def non_start(a, b):
  '''(str, str) -> str
  Return a+b, but without the first characters of a and b.
  >>> non_start('Hello', 'There')
  'ellohere'
  >>> non_start('java', 'code')
  'avaode'
  '''

  # omit the first character of a
  first = a[1:]
  # omit the first character of b
  second = b[1:]

  # concatenate the new strings and return
  return first+second
```

Q1 Sudeep has written a function that, given HTML tag, returns a word surrounded with the HTML tag. Add in subgoal labels for his code:

```
def make_tags(tag, word):
  '''
  (str, str) -> str
  Return word, surrounded with HTML-style tag around it.
  >>> make_tags('i', 'Yay')
  '<i>Yay</i>'
  >>> make_tags('cite', 'Hello')
  '<cite>Hello</cite>'
  '''


  # LABEL 1: ____________________________________
  open_tag = '<' + tag + '>'


  # LABEL 2: ____________________________________
  close_tag = '</' + tag + '>'


  # LABEL 3: ____________________________________
  return open_tag + word + close_tag
```

Function Design Recipe and Strings

1. Mohammed is writing a function that, given a string name, returns a greeting of the form “Hello *(name)*!”. He started writing the function with the design recipe. Finish the function for him using the function design recipe.

```
>>> hello_name('Bob')
'Hello Bob!'
>>> hello_name('Alice')
'Hello Alice!'
>>> hello_name('X')
'Hello X!'
```

2. Using the function design recipe, write a function that, given a string of even length, returns the first half. So the string “WooHoo” yields “Woo”.

(All questions on this worksheet are adapted from Nick Parlante’s CodingBat exercises for strings.)