CSC108H Worksheet: String Operations

1. Consider this code:

   phrase = 'laughing out loud'

   Assuming the code above has been executed, circle the expression(s) that will produce 'lol'.
   (a) phrase[0] + phrase[9] + phrase[0]
   (b) phrase[0] + phrase[9] + phrase[-4]
   (c) phrase[-4] + phrase[-3] + phrase[0]

2. Consider this code:

   a = 'scores'
   b = a[2:]

   Assuming the code above has been executed, circle the expression(s) that will produce True.
   (a) a == 'scores' and b == 'scores'  
   (b) a == 'ores' and b == 'ores'  
   (c) a == 'cores' and b == 'scores'  
   (d) a == 'cores' and b == 'cores'  
   (e) a == 'scores' and b == 'ores'

3. Consider this code:

   lyrics = 'abc easy as 123'

   Assuming the code above has been executed, circle the expression(s) that produce False.
   (a) 'easy' in lyrics 
   (b) str(len('mj')) in lyrics 
   (c) 'cab' in lyrics 
   (d) '' in lyrics

4. Consider this code:

   s = 'Jacqueline'

   You know that the slicing operation s[1:4] will produce the string 'acq'. The slicing operation has an optional third parameter that determines the stride (or distance between characters) in the slice. For example, the slicing operation s[::2] will produce the string 'Jculn', which has every other character in 'Jacqueline', starting from the first character in the string, and up to the end of the string. Use a negative stride to work backwards through a string.

   (a) Write an expression that uses slicing on s to produce the string 'aqeie'.
   (b) Write an expression that uses slicing on s to produce the string 'enileuqcaJ'.
   (c) Write an expression that uses slicing on s to produce the string 'eieqa'.