CSCA08H Worksheet: String Operations

1. Consider this code:

```
Laughing  Out   Loud
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
-17              -8          -4
```

```
phrase = 'Laughing Out Loud'
```

Assuming the code above has been executed, complete the indices in the expression below that will produce the string `'LOL'`. Show two different solutions!

```
phrase[ 0  ] + phrase[ 9  ] + phrase[ 13 ]

phrase[ -4 ] + phrase[ -8 ] + phrase[ -17 ]
```

2. Consider this code:

```
big orange  cat
0 1 2 3 4 5 6 7 8 9 10 11 12 13
                    -4 -3 -2 -1
```

```
phrase = 'big orange cat'
slice1 = phrase[:3]
slice2 = phrase[-4:]
slice3 = phrase[3:8]
```

Assuming the code above has been executed, complete the table with the values that each variable refers to.

Variable	Value
`phrase`	'big orange cat'
`slice1`	'big' (character at index 3 is excluded)
`slice2`	' cat' (first character is space)
`slice3`	' oran' (first character is space; character at index 3 included, at index 8 excluded)

3. Consider this code:

```
lyrics = 'abc easy as 123'
```

Assuming the code above has been executed, circle the expression(s) that produce `False`.

T (a) `'easy' in lyrics` T (b) `str(len('mj')) in lyrics`

F (c) `'cab' in lyrics` T (d) `'' in lyrics`

len('mj') evaluates to 2 of type int
str(len('mj')) evaluates to '2' of type str
'2' in lyrics evaluates to True

4. Consider this code:

```
s = 'csca08 is great!'
     0     6  9     15
```

You know that the slicing operation `s[3:6]` will produce the string `'a08'`. 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 `'cc0 sget'`, which has every other character in `'csca08 is great!'`, 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 `'csca08'`.

s[:6]

(b) Write an expression that uses slicing on `s` to produce the string `'s0ig'`.

s[1:11:3] indices 1, 4, 7, 10

(c) Write an expression that uses slicing on `s` to produce the string `'!ar i8a'`.

s[:-14:-2] indices -1, -3, -5, -7, -9, -11, -13