

1. Suppose string `s = "DATA STRUCTURES"`. Evaluate the following expressions:

`s[0:3] + s[-1]` **'DATS'**

`s[5:9] + s[:2]` **'STRU DA'**

2. Consider the function `mystery1`:

```
1 def mystery1(word):  
2     n = len(word) // 2  
3     return word[n]
```

Given the function call `mystery1("PYTHON")`,

a) What is the value of `n` after line 2 is run? **3**

b) What value is returned? **'H'**

3. Consider the function `mystery2`:

```
1 def mystery2(s):  
2     n = len(s) // 2  
3     return s[n:] + s[:n]
```

What is returned by `mystery2("keyboard")`? **'oardkeyb'**

4. What is the range of possible values for each of the following? Give the smallest and largest possible value.

a) `random.randint(3, 7)` **3 to 7 (both included)**

b) `random.random()` **0.0 up to, but not including, 1.0**

c) `random.randint(-5, 5)` **-5 to 5 (both included)**

5. Fill in the blanks so that `roll_two_dice()` returns the sum of two six-sided dice (each die is a random integer from 1 to 6):

```
1 import random  
2 def roll_two_dice():  
3     die1 = random.randint(1, 6)  
4     die2 = random.randint(1, 6)  
5     return die1 + die2
```

a) Smallest possible return value: **2**

b) Largest possible return value: **12**

6. Draw the complete call stack trace for this program: a frame for every function call, in the order created, from the initial call through to the point where answer receives its final value. For each frame, fill in its parameter values and, once known, the value it returns to its caller. Use the boxes below, one per frame.

```
def double(n):  
    return n * 2  
  
def combine(a, b):  
    return double(a) + double(b)  
  
answer = combine(3, 4)
```

main
answer: ? → 14

combine
a: 3 b: 4
returns double(3) + double(4)
 → 6 + 8 → 14

double - first call
n: 3
returns 3 * 2 → 6

double - second call
n: 4
returns 4 * 2 → 8