Q. Which of the following output could NOT be produced by the code below?

```python
import random
var1 = random.randrange(19, 41, 4)
var2 = random.randrange(17, 45, 5)
var3 = random.randrange(42, 18, -3)
print(max(var1, var2, var3))
```

(a) 21
(b) 32
(c) 42
(d) 39
(e) 17

Q. What is the output produced by the following code?

```python
import random
var1 = random.randrange(19, 41, 4)
var2 = random.randrange(17, 45, 5)
var3 = random.randrange(42, 18, -3)
print(max(var1, var2, var3))
value = 3 ** (1 + 1 * 2) // 5 % (20 / 2)
print(value)
```

(a) 21
(b) 32
(c) 42
(d) 39
(e) 17

Q. What is the output produced by the following main() function?

```python
def main():
    a_list1 = [5, 3, 2]
    a_list2 = [4, 6]
    fiddle_lists(a_list1, a_list2)
    print("a_list1:", a_list1)
    print("a_list2:", a_list2)

def fiddle_lists(list1, list2):
    list1 = [1, 5]
    list2.append(list1[1])
    list1.append(list2[0])
    list1.append(list2[1])
```

(a) a_list1: [1, 5, 4, 6]
a_list2: [4, 6]
(b) a_list1: [5, 3, 2, 4, 6]
a_list2: [4, 6]
(c) a_list1: [1, 5, 4, 6]
a_list2: [4, 6, 5]
(d) a_list1: [5, 3, 2]
a_list2: [4, 6]
(e) a_list1: [5, 3, 2]
a_list2: [4, 6, 5]

Q. Given the following code, what is the type of each of the three Python objects (object1, object2 and object3)? (From SS, 2017)

```python
a_tuple = (3, 4, "five")
a_list = [2, 'one', 5, 4]
a_dict = {'a': 'bdegh', 'b': '135'}
```

object1: (a) a_tuple: (3, 4, "five")
(b) a_list: [2, 'one', 5, 4]
(c) a_dict: {'a': 'bdegh', 'b': '135'}

Q. Complete the output produced by the main() function? (From SS, 2017)

```python
def main():
    list1 = [3, 2, 1]
    fiddle2(list1)
    print("List1:", list1)

def fiddle2(list1):
    list2 = []
    list2.append(list1[0])
    list2.append(list1[1])
    list1 = list2
    list1[0] = list2[0] + 3
```

List1:

```
(a) List1: [4, 5, 1]
(b) List1: [3, 2, 1]
(c) List1: [3, 1, 1]
(d) List1: [3, 2, 1]
(e) List1: [3, 1, 1]
```

Q. Complete the convert_first_letter() function which is passed a list of names as a parameter. The function changes the first letter of each name in the list to uppercase, leaving the rest of the name unchanged. You can assume that each element of the list contains at least one character. For example, executing the following program with the completed function, prints:

1. names: ['karl', 'Orlando', 'carlo', 'zAC']
2. names: ['Karl', 'Orlando', 'Carlo', 'ZAC']

```python
def main():
    names = ['karl', 'Orlando', 'carlo', 'zAC']
    print("1. names:", names)
    convert_first_letter(names)

    names = ['Karl', 'Orlando', 'Carlo', 'ZAC']
    print("2. names:", names)
    convert_first_letter(names)
```

```python
def convert_first_letter(names_list):
    for index in range(len(names_list)):
        names_list[index] = names_list[index][0].upper() + names_list[index][1:]
```

Q. In the docstring of the do_a_check() function, add ONE doctest which does not fail. (From SS, 2017)

```python
def do_a_check(value1, value2, value3):
    """Checks the parameter values function, add ONE doctest which does not fail. (From SS, 2017)"
    a_list = [value1, value2, value3]
    a_list.sort()
    return a_list[0] == value1 and a_list[1] == value2
```

```python
import doctest
doctest.testmod()
```
### Dictionaries - 2016 S1 Question 18 a) (6 marks)

Complete the `get_indexes()` function which takes two parameters: 
- `sentence`: a string of words separated by spaces (no punctuation or capitals)
- `target`: a target word

The function returns a list containing the index positions of the target word in the sentence (starting from index 0).

```python
def get_indexes(sentence, target):
    result = []
    return result
```

```python
def main():
    word = 'equal'
    message = "all animals are equal but some animals are more equal than others"
    result = get_indexes(message, word)
    print(word, "is at indices:", result)
main()
```

### Dictionaries - 2016 S1 Question 18 b) (7 marks)

Complete the `build_index_dict()` function which takes a string as a parameter and returns a dictionary. The key of each dictionary item is a target word (each word of the parameter string) and the value of each dictionary item is the list of index positions of all occurrences of the target word in the parameter string. Your code must call the `get_indexes()` function.

```python
def build_index_dict(sentence):
    def get_indexes(sentence, target):
        ... # See previous slide
    def main():
        message = "all animals are equal but some animals are more equal than others"
        my_dict = build_index_dict(message)
        for word, indices in my_dict.items():
            print(word, indices)
    main()
```

### Nested loops, GUI's – 2016 S1 Question 19 (13 marks)

```python
from tkinter import *
def rectangular_grid(a_canvas):
    number_of_columns = 3
    number_of_rows = 5
    left_hand_side = 10
    y_down = 10
    size = 10
    dist = size * 3
    for i in range(number_of_rows):
        if i % 2 == 0:
            x_left = left_hand_side
        else:
            x_left = left_hand_side + size
        for j in range(number_of_columns):
            rect = (x_left, y_down, x_left + size, y_down + size)
            a_canvas.create_rectangle(rect, fill='blue')
            x_left = x_left + dist  # Position A
            y_down = y_down + size  # Position B
```