https://bit.ly/33J0Gic

https://bit.ly/2LpO9tw ANSWERS

1. What does the following program do:

```
def flup(list):
    if len(list) == 1:
        return list[0]
    else:
        return list[0] + flup(list[1:])

print(flup([1,3,5,7,9]))
```

   a. Returns the sum of every element in a list excluding the first one
   b. Returns the sum of every element in a list
   c. Iterates through a list in reverse and computes the sum of every other element
   d. Nothing, this program crashes

2. What is the iterative equivalent to the following program?

   a.
```
def listsum(numList):
    theSum = 0
    for i in range(len(numList)):
        theSum = theSum + numList[i]
    return theSum
```

   b.
```
def listsum(numList):
    theSum = 0
    i = 0
    while i <= len(numList):
        theSum = theSum + numList[i]
        i += 1
    return theSum
```

   c.
```
def listsum(numList):
    theSum = 0
    i = 0
    while i < len(numList):
        theSum = theSum + numList[i]
        i += 1
    return theSum
```

   d. None of the above, this problem cannot be solved iteratively
   e. Both a and c are correct

https://bit.ly/33J0Gic

https://bit.ly/2LpO9tw ANSWERS

3. Consider the following classes

```
class Person(object):
    def __init__(self, firstname, lastname):
        self.firstname = firstname
        self.lastname = lastname
    def __eq__(self, other):
        return self.firstname == other.firstname and
        self.lastname == other.lastname
    def __str__(self):
        return '(self.lastname, self.firstname)'
        ---  ---  ---  ---  ---  ---  ---  ---
class Musician(Person):
    def __init__(self, firstname, lastname, instrument):
        super().__init__(firstname, lastname)
        self.instrument = instrument
```

What does the following program print?

```
p1 = Person('Squid', 'Ward')
p2 = Musician('Squid', 'Ward', 'clarinet')
print(p1 == p2, end=" ")
print(p1 is p2)
```

   a. True True
   b. True False
   c. False True
   d. False False

4. What is the approximate runtime of the following function?

```
def OhBigNo(L):
    '''(list de int)->None"
    n=len(L)
    for i in range(n):
        for j in range(25):
            L[i] = L[i] + 10
    for i in range(n):
        print(L[i])
```

   a. $O(Log_2n)$
   b. $O(n)$
   c. $O(nlog_2n)$

https://bit.ly/33J0Gic

https://bit.ly/2LpO9tw ANSWERS

d. $O(n^2)$
e. $O(n^3)$

5. Consider the following program

```
class Point:

    def __init__(self, initX, initY):
        self.x = initX
        self.y = initY

    def __str__(self):
        return "x=" + str(self.x) + ", y=" + str(self.y)
class LabeledPoint(Point):

    def __init__(self, initX, initY, label):
        super().__init__(initX, initY)
        self.label = label

    def __str__(self):
        return super().__str__() + " (" + self.label + ")"
```

What does the following print?

```
p = LabeledPoint(7, 6, "Here")
p2 = Point(6, 7)
print(p)
```

a. x = 7, y = 6 (Here)
b. x = 7, y = 6 **and crashes**
c. x = 6, y = 7 (Here)
d. x = 6, y = 7 **and crashes**

https://bit.ly/33J0Gic

https://bit.ly/2LpO9tw ANSWERS

6. Does the following program's docstring describe its functionality properly for all input numbers >= 1?

```
def premier(n):
    '''(int)->bool
    Returns True is the n input is a prime number and False
    otherwise.
    Precondition: n is a positive integer greater than 1'''
    is_prime = True
    i = 2
    while i < n:
        if n % i == 0:
            is_prime = False
        i = i + 1
    return is_prime
```

   a. Yes
   b. No its false for n = 2
   c. No it can be corrected by adding the following code after the `is_prime = False` affectation

```
else :
    is_prime = True
```

   d. No, it can be corrected by replacing `while i < n` with `while i <= n`.