Time to put your knowledge to the test!

But instead of grades, you get prizes!

Let’s start with a simple question!

1) What would the output of this python code be?

```
type(“True”)
```

A	B
<class 'str'>	<class ‘bool’>
C	**D**
<class ‘int’>	It would throw an error

Onto the next one!

2) Which of these code blocks removes all multiples of 10 between 0 and 100 (inclusive) from a set?

A	B

`s = {... elements of original set}`
 	
`r = set()`
`for i in range(10, 100+1, 10):`
`    r.add(i)`
 	
`s.difference_update(r)`
	`s = {... elements of original set}`
	
	`r = set()`
	`for i in range(10, 100, 10):`
	`    r.add(i)`
	
	`s.union(r)`

C	**D**

`s = {... elements of original set}`
 	
`r = set()`
`for i in range(10, 100+1, 10):`
`    r.add(i)`
 	
`s.intersection(r)`
	`s = {... elements of original set}`
	
	`r = set()`
	`for i in range(10, 100, 10):`
	`    r.add(i)`
	
	`s.difference(r)`

Onto the next one!

3) What is the time complexity of this algorithm?

```
def mypow3(x, n):
    if n == 0:
        return 1
    else:
        res = mypow2(x, n//2)
        if n %2 == 0: # testing if n is even number
            return res*res
        else:
          return x*res*res
```

A	B
O(n)	O(log n)
C	**D**
O(n^2)	O(n^3)

Onto the next one!

4) Same algorithm as before, what is the maximum depth reached?

```
def mypow3(x, n):
    if n == 0:
        return 1
    else:
        res = mypow2(x, n//2)
        if n %2 == 0: # testing if n is even number
            return res*res
        else:
          return x*res*res

mypow3(2, 50)
```

A	B
n	n + 1
C	**D**
log(n)	n log(n)

Onto the next one!

5) Given a non-empty list of integers “nums”, every element appears twice, apart from only one element.

Which of these options can find this element?

Example:

Input: [3, 2, 5, 6, 6, 2, 3]
Output: 5

Because every number apart from 5 appears twice!

A	B
1. Iterate over all the elements in nums 2. If some number in nums is new to array, append it to list *seen* 3. If some number is already in the array, remove it from *seen* 4. Once the iteration is complete, return *seen[0]*	1. Iterate over all the elements in nums 2. If some number in nums is new to array, remove it from list *seen* 3. If some number is already in the array, append it to list *seen* 4. Once the iteration is complete, return *seen[0]*
C	**D**
1. Iterate over all the elements in nums 2. Check if one number is equal to the next 3. Return first number that is different from its next number	1. Sort the list 2. Iterate over list and check if one number is equal to the next 3. Return first number that is different from its next number and its previous number

Onto the next one!

6) What is the Big O for this very fancy function

```
def fancyFunction(l):
    X = sorted(l)

    if len(X)==1:
        return True

    for i in range(len(X)):
        if i==0 and X[i]!=X[i+1]:
            return True

        elif i==len(l)-1 and X[i]!=X[i-1]:
            return True

        elif i!=0 and i!=len(l)-1:
            if X[i]!=X[i-1] and X[i]!=X[i+1]:
                return True

    return False
```

A	B
O(n)	O(log n)
C	**D**
O(n log n)	O(n^2)

Onto the next one!

7) What is the printed output of this code?

```
class CardGame():
        colors = (‘Heart’, ‘Diamond’, ‘Spade’, ‘Club’)
        vals = (2,3,4,5,6,7,8,9,10,’J’,’Q’,’K’,’A’)
        def __init__(self):
                self.cards = []
                for color in range(4):
                        for val in range(13):
                                self.cards.append((val, color))

game = CardGame()
print(game.colors)
```

A	B
(‘Heart’, ‘Diamond’, ‘Spade’, ‘Club’)	(‘Heart’)
C	**D**
(2,3,4,5,6,7,8,9, ’J’,’Q’,’K’,’A’)	Error Message

Onto the next one!

8) How would you rename key ‘A’ to key “new_A” in the dictionary:

```
dict = {
  "A": "1",
  "B": 2,
  "C": 3,
  "D": "4"
}
```

A	B
dict['A'] = dict.pop('new_A')	dict['A'] = dict.pop('new_A')
C	**D**
dict['new_A'] = dict.pop('A')	dict['new_A'] = dict.pop('A')

Onto the next one!

9) What will this print?

```
dict = {'a':1, 'b':2}
print(dict['c'])
```

A	B
None	0
C	**D**
KeyError	Infinite loop

Onto the next one!

10) Given the function:

```
def fib(n):
        If n < 2: return 1
        return fib(n-2) + fib(n-1)
```

What is the time complexity?

A	B
O(2*n)	O(log n)
C	**D**
O(n^2)	O(2^n)

Onto the next one!

11) Given an array of numbers nums, and an integer target, return two numbers from nums such that they add up to target.

Example:

Input:

nums: [3, 2, 4]
target: 6

Output:

[1,2]

because nums[1] + nums[2] = target

A

1. Create loop i that loops through each element starting at index 0
2. Create another loop j that starts at index 1
3. Check if nums[i]+nums[j]=target
4. If yes, return i and j, if not, keep going

B

1. Create loop i that loops through each element starting at index 0
2. Create another loop j that starts at index 1
3. Check if i==j. If yes, skip step 4.
4. Check if nums[i]+nums[j]=target
5. If yes, return i and j, if not, keep going

C

1. Dictionary is initially empty. Put first element of array in it as a key with value the index.
2. Look at next element *i* and check if the result of target minus *i* is in the dictionary.
3. If match found, return the pair. If not, put it in as key *i* with value index.
4. Keep going until find solution, and return the value of key and index of element.

D

1. Create a loop i that loops through each element starting at index 0
2. Each time, have a variable *X* = target - nums[i]
3. Use binary search to find this *X*.
4. If it exists, return i and index of *X*. If not, keep looping.

That’s it!

A	B
C	D