

Final Quiz

Name:

We would also love to know more about you, and to keep in touch after the course. We encourage you to join the "AddisCoder 2016" Facebook group, which you can find via a Facebook search. Also feel free to leave some further information about yourself below.

E-mail:

Phone number:

City/Town:

High school:

Please answer the questions according to the following example.

Example

What will be the output of the code below?

```
In [ ]: def addOne(x):  
        return x + 1  
  
        print addOne(2)
```

Answer

3

Problem 1

What will be the value of x?

```
In [ ]: x = 2**3-2*3
```

Problem 2

What will be the output of the code below? If there's an error, explain why.

```
In [2]: L = [1, 2, 3]
        i = 0
        while i < len(L):
            print i
```

Problem 3

What will be the output of the code below? If there's an error, explain why.

```
In [ ]: L = [2, 1, 0]
        for x in L:
            print L[x]
```

Problem 4

What will be the output of the code below? If there's an error, explain why.

```
In [5]: L = '210'
        for x in L:
            print L[x]
```

Problem 5

What will be the output of the code below? If there's an error, explain why.

```
In [ ]: def multiply(x,y):
        return x + multiply(x, y-1)
        print multiply(5, 4)
```

Problem 6

Suppose someone has given you an implementation of `solve2` such that `solve2(a,b,c,d,e,f)` returns a list `L` containing the solution to the pair of equations $ax + by + c = 0$ and $dx + ey + f = 0$. If there is no unique solution, `solve2` returns `None`. Otherwise, `L[0]` is the solution for x , and `L[1]` for y . Which of the following is a correct implementation of `solve(a,b)`, which returns a solution to $ax + b = 0$?

```
In [ ]: # A
def solve(a,b):
    return solve2(a,b,0,0,0,0)

# B
def solve(a,b):
    return solve2(a,0,b,0,0,0)[0]

# C
def solve(a,b):
    return solve2(a,b,0,1,0,0)[0]

# D
def solve(a,b):
    return solve2(a,0,b,1,1,0)[0]
```

Problem 7

What will be the output of the code below? If there is an error, explain why

```
In [ ]: def mystery(x):
        return 2*x + 7
        print mystery('10')
```

Problem 8

What will be the output of the code below? If there is an error, explain why

```
In [7]: def mystery(x):
        return 3*x
        print mystery('7')
```

Problem 9

Consider the algorithm you learned for multiplication when you were a young child, based on memorizing your multiplication tables for all numbers between 0 and 9. Let's say for multiplying numbers with n digits, it took about $4n^2$ steps. Suppose you spent some more time and now memorized multiplication tables for all numbers between 0 and 99. With this increased memorization, now how many steps would the grade school algorithm take?

- A. $4n^2$
- B. $2n^2$
- C. n^2
- D. $4n$

Problem 10

Suppose I wish to write a function `findIt(x, L)` which takes as input a number `x` and an arbitrary list `L` and returns any index `i` such that `L[i] == x`. If `x` doesn't appear in `L`, the function should return `-1`. Which algorithmic concept should I use to implement `findIt`? Circle any that apply.

- A. for loop
- B. binary search
- C. hashing
- D. while loop

Problem 11

Suppose we want to implement a function `foo(x, L)` in which `L` is sorted. This function should return `True` if `foo` is in `L` and return `False` otherwise. `x` is an `int` and `L` is a list of `ints`. What error might the following code give if run?

```
In [15]: # check if x is in L[A:B]
def binarySearch(x, L, A, B):
    if B < A+1:
        return False
    else:
        mid = (A+B)/2
        if L[mid] == x:
            return True
        elif L[mid] > x:
            return binarySearch(x, L, A, mid)
        else:
            return binarySearch(x, L, mid, B)

def foo(x, L):
    return binarySearch(x, L, 0, len(L))
```

- A. maximum recursion depth exceeded
- B. index out of range error
- C. No error will be printed by Python, the code is correct.
- D. No error will be printed by Python, but there is a mistake in the code so sometimes it gives the wrong answer.

Problem 13

In the mobile application Snapchat, users post pictures and videos from their phone. Then other users can "follow" them to see everything they post. If you follow me, I don't have to follow you back. We want to make a graph representing who is following who on Snapchat. What would the most natural representation be?

- A. Undirected graph
- B. Directed graph
- C. Either of the above is fine
- D. This is not a proper situation to use a graph representation

Problem 14

Jelani's friend is named Tinchel, and Tinchel believes that memoization should always be used in recursive situations to make things faster. For example, MergeSort should be memoized to make it faster. Is Tinchel right?

- A. Yes, memoization always makes recursive algorithms faster.
- B. No, memoization doesn't always help, but it would make MergeSort faster.
- C. No, memoization doesn't always help. For MergeSort it wouldn't help since it never recurses on the same parameters twice.
- D. Yes, memoization should always be used, but Tinchel's reason is wrong. The real reason is that it uses less memory.

Problem 15

What would be the running time of integer multiplication if we had a way of breaking the input integers into 4 pieces and doing 5 recursive calls, then combining the results in $O(n)$ time? Here n is the number of digits.

A. $O(n^{\log_2 3})$

B. $O(n^{\log_2 5})$

C. $O(n^{\frac{1}{2} \cdot \log_2 5})$

D. $O(n)$