

CSCI 150: Practice Exam 2

Read all questions carefully before beginning. You will have 1 hour to complete this exam. You are not allowed to use your notes, textbook, phone or computer. Show all your work for full credit.

1. (20 points) Sam is tasked with writing a function that takes a list of numbers as input and returns `True` if every even number in the list is adjacent to an odd number, and `False` otherwise. He writes the following incorrect code:

```
def adjacent(nums):
    for i in nums:
        if nums[i] % 2 == 0:
            if nums[i - 1] % 2 == 1 or nums[i + 2] % 2 == 1:
                return True
            else:
                return False
```

Describe three things that are wrong with his code and how you would fix it.

2. (20 points) What is the value of `n` after execution of the following Python program? **Show your work for full credit.**

```
t = "WHEREINTHEWORLDIS"  
r = "CarmenSandiego"  
p = t.find("R") * 4 - 1  
f = r[-3:]  
n = t[p:p + 2] + f + "N"  
n.lower()
```

3. (30 points) We discussed the Collatz conjecture earlier in the semester. Here is a similar process for an arbitrary positive integer n :

- If n is a multiple of three, divide it by three.
- Otherwise, double it and add one.

For example, starting with $n = 10$, one gets the sequence 10, 21, 7, 15, 5, *etc.*

Using a while loop, write a function in Python that accepts a number n as a parameter, and repeats the process above until either $n = 1$ or until 100 steps are reached. The function should return how many steps the process took to reach 1, or return -1 if the process went on for 100 steps.

```
def splaz(n):
```

4. (30 points) Write a function called `isogram` using a **for loop** that takes in a string and returns `True` if there are no repeated letters in the word, and `False` otherwise. It does not matter whether the repeated letters are next to each other. For example:

- `isogram("lycanthropies")` returns `True`
- `isogram("palindrome")` returns `True`
- `isogram("mississippi")` returns `False`
- `isogram("abecedarian")` returns `False`

```
def isogram(s):
```