COSC 101, Exam #2
March 2020

Name: _____

Please write your name above. Do not start the exam until instructed to do so.

There are 5 questions and a total of 48 points available for this exam. Don't spend too much time on any one question.

Since indentation is important in Python, please be sure that your use of indentation is obvious for any code you write.

If you want partial credit, show as much of your work and thought process as possible.

If you run out of space for answering a question, you can continue your answer on one of the blank pages at the end of the exam. If you do so, be sure to indicate this in two places: (1) below the question, indicate which blank page contains your answer, and (2) on the blank page, indicate which question you are answering.

Question	Points	Score
1	8	
2	11	
3	6	
4	10	
5	13	
Total:	48	

1. (8 points) Assume that the following statements have already been executed:

```
a = "Super"
b = 'tuesday'
c = 2020
d = 2
```

For each of the following expressions, evaluate the expression and write the resulting value, or identify the error in the code that would prevent it from running.

(a) `a[0] in b`

Solution:

```
False
```

(b) `len(a) / d`

Solution:

```
2.5
```

(c) `str(c)[1:3:1]`

Solution:

```
'02'
```

(d) `a[len(a) % d] + str(d)`

Solution:

```
'u2'
```

(e) `b[0] = 'T'`

Solution:

```
Error: 'str' object does not support item assignment
```

(f) `not ('E' in a.upper() and 'a' not in b)`

Solution:

```
True
```

(g) `a[len(b)]`

Solution:

```
Error: string index out of range
```

(h) `d // len(a) + c`

Solution:

```
2020
```

2. (a) (5 points) What is the output of the following program?

```
def results(x, y):
    if x == y:
        print(x, "and", y, "are the same")
    elif x < y:
        print(x, "is smaller than", y)
    else:
        print(x, "is bigger than", y)

def mystery(x):
    if x < 3:
        y = x
    elif x > 10:
        y = 10
    else:
        y = 5
    results(x, y)

x = 1
y = 2
results(x, y)
mystery(y)
results(x,y)
```

Solution:

```
1 is smaller than 2
2 and 2 are the same
1 is smaller than 2
```

(b) (6 points) What is the output of the following program if the user enters today's date (11)?

```
def is_it_break(day):
    days_until_break = break_day - day
    while days_until_break > 0:
        print("March " + str(day) + ": " + str(days_until_break) +\
              " days until spring break.")
        day +=1
        days_until_break = break_day - day
    print("March " + str(day) + ": SPRING BREAK!")

break_day = 14
date = int(input("What date is it? "))
is_it_break(date)
```

Solution:

```
What date is it? 11
March 11: 3 days until spring break.
March 12: 2 days until spring break.
March 13: 1 days until spring break.
March 14: SPRING BREAK!
```

3. (6 points) Write a function named `to_singular` that accepts an English plural noun as a parameter (a string) and returns the singular of that noun (also a string). For simplicity, you are only required to handle two *de-pluralization* cases:

   1. If the noun ends in `ies`, remove the `ies` and replace with a `y`. For example, `to_singular("puppies")` should return `"puppy"`, and `to_singular("cities")` should return `"city"`.
   2. If the noun ends in `s`, remove the `s`. For example, `to_singular("chefs")` should return `"chef"`, and `to_singular("roofs")` should return `"roof"`.
   3. If the noun ends in anything other than `ies` or `s`, return the string unchanged. For example, `to_singular("moose")` should return `"moose"`.

   You may not use any string methods for this problem. String slicing is fine to use.

Solution:

```
def to_singular(s):
    if s[-3:] == 'ies':
        return s[:-3] + 'y'
    elif s[-1] == 's':
        return s[:-1]
    else:
        return s
```

4. (10 points) You are part of a running group with a race coming up. Write a program to help the runners in your group keep track of their training. Your program should ask how many days away the race is, then ask how many miles the runner ran each day, then print out a summary of their training.

   Your program must use a while loop. And the output of your program must exactly match these examples:

```
How many days until your race? 3
How many miles did you run on day 1? 3.1
How many miles did you run on day 2? 0
How many miles did you run on day 3? 6.2
You trained by running 9.3 miles, averaging 3.1 miles per day

How many days until your race? 1
How many miles did you run on day 1? 0
You didn't do any training for your race...
```

Solution:

```
start_days = int(input("How many days until your race? "))
days_until_race = start_days
total_miles = 0
day = 1

for _ in range(days_until_race):
    total_miles += float(input("How many miles did you run on day " + str(day) + "? "
    days_until_race  = 1
    day += 1

if total_miles == 0:
    print("You didn't do any training for your race...")
else:
    print("You trained by running", total_miles, "miles, averaging", total_miles / st
```

5. A team of scientists studying COVID-19 need to locate genes within DNA sequences. The sequences and genes are represented as strings containing the letters A, T, C, G.

   (a) (8 points) Write a function that finds the *start* of the first occurrence (if any) of a gene in a DNA sequence. If the gene does not exist in the DNA sequence, then the function should return -1. Here are some examples:

DNA sequence	Gene	Start
ATCGATAT	CGA	2
TCGATCGA	CGA	1
ACTGATAT	CGA	-1

Select one line of code from each of the pairs of lines of code below and reorder them to create a function that solves the problem.

```
A1          if sequence[i] != gene[j]:
A2          if sequence[i+j] != gene[j]:

B1          return j
B2          return i

C1        match = True
C2        match = False

D1        if match:
D2        if not match:

E1    return len(sequence) - len(gene)
E2    return -1

F1    for i in range(len(sequence) - len(gene) + 1):
F2    for i in range(len(sequence)):

G1              return -1
G2              match = False

H1        for j in gene:
H2        for j in range(len(gene)):
```

Select only 8 lines of code from above, and only one line from each pair. You may fill in line identifiers (*e.g.*, E2) below, or write out the code.

```
def find_gene_start(sequence, gene):
```

Solution:

- F 1
- C 1
- H 2
- A 2
- G 2
- D 1
- B 2
- E 2

(b) (5 points) Now write a function that finds the *end* of the first occurrence (if any) of a gene in the DNA sequence. If the gene does not exist in the DNA sequence, then the function should return -1. Here are some examples:

DNA sequence	Gene	End
ATCGATAT	CGA	4
TCGATCGA	CGA	3
ACTGATAT	CGA	-1

Your function *must use the previous function* that finds the start of the first occurrence (if any) of a gene in the DNA sequence. You may assume the previous function works correctly as described (regardless of whether of not your particular solution is correct).

Solution:

```
def find_gene_end(sequence, gene):
    start = find_gene_start(sequence, gene)
    if start ==  1:
        return  1
    return start + len(gene)   1
```

(This page is intentionally blank. Label any work with the corresponding problem number.)

(This page is intentionally blank. Label any work with the corresponding problem number.)