CS128 Lab Final Exam Spring 2022

Programming Task Guidelines:

1. Close all browser tabs except for codio.com, pythontutor.com, zybooks.com, and canvas.jcu.edu. You are not allowed to open any other website.
2. You may refer to the code examples from the lessons, labs, and homework.
3. You may refer to printed or handwritten notes while working on the programming tasks.
4. You are not allowed to communicate with anyone except the instructor.
5. When you are finished, save your notebook and submit lab_final.pdf and lab_final.ipynb to Canvas.

Task1a (4 points):

You need to save $25.

Each day you save a random amount between $5 and $9 (inclusive).

Loop to show the daily savings and amount remaining to reach the goal.

Stop looping if goal is met or exceeded.

You may have to run several times to see the goal met without being exceeded.

SAMPLE OUTPUT:

```
I need to save 25 dollars
Saved $9 on day 1, need to save $16 more
Saved $5 on day 2, need to save $11 more
Saved $5 on day 3, need to save $6 more
Saved $5 on day 4, need to save $1 more
Saved $9 on day 5, goal exceeded by $8!
```

SAMPLE OUTPUT:

```
I need to save 25 dollars
Saved $7 on day 1, need to save $18 more
Saved $6 on day 2, need to save $12 more
Saved $5 on day 3, need to save $7 more
Saved $7 on day 4, goal met!
```

Task1b (1 points):

Copy the code from task1a into task1b.

Update the code to require $25 to be saved in 4 days or less.

Note it is possible to save $25 or more in fewer than 4 days.

The loop should stop if the goal of $25 is reached or if more than 4 days has passed.

Indicate if the goal was not met after 4 days.

SAMPLE OUTPUT:

```
I need to save 25 dollars in 4 days or less
Saved $5 on day 1, need to save $20 more
Saved $7 on day 2, need to save $13 more
Saved $7 on day 3, need to save $6 more
Saved $5 on day 4, need to save $1 more
Goal not met
```

SAMPLE OUTPUT:

```
I need to save 25 dollars in 4 days or less
Saved $8 on day 1, need to save $17 more
Saved $7 on day 2, need to save $10 more
Saved $8 on day 3, need to save $2 more
Saved $6 on day 4, goal exceeded by $4!
```

SAMPLE OUTPUT:

```
I need to save 25 dollars in 4 days or less
Saved $6 on day 1, need to save $19 more
Saved $5 on day 2, need to save $14 more
Saved $8 on day 3, need to save $6 more
Saved $6 on day 4, goal met!
```

SAMPLE OUTPUT:

```
I need to save 25 dollars in 4 days or less
Saved $8 on day 1, need to save $17 more
Saved $8 on day 2, need to save $9 more
Saved $9 on day 3, goal met!
```

Task2a (3 points):

Create a function named **exactly_2_match** that takes 3 parameters.

The function should return **True** if exactly 2 out of the 3 parameters have the same value, otherwise **False**. The function should return **False** if all 3 parameters have the same value.

Implement the function without using nested or chained conditional statements, you can use the boolean operators **and**, **or**, **not**.

Test your function with the sample test cases.

Task2b (3 points):

Create a function named **exactly_2_match_V2** that implements the same functionality but does not use the operators **and**, **or**, **not**.

You must use nested and/or chained conditional statements to achieve the functionality.

Task3 (5 points):

To define a “function header” in Python:

- begin with the keyword **def** followed by a space
- followed by the function name
- followed by a pair of parentheses which may or may not contain parameters
- followed by a colon

For example, the code below shows the function header for a function named “dostuff”:

def dostuff(a,b,c):

Write a function **extract_function_name** that takes a string representing a function header as a parameter.

The **extract_function_name** function should extract the substring that represents the name from the parameter string.

The **extract_function_name** function should return the empty string if the parameter string does not start with the keyword **def** followed by a space or if the parameter string does not have a left parenthesis following the def keyword to indicate the end of the function name.

Don’t worry about checking for the right parenthesis or colon.

Task4 (4 points)

Write a function **review_lookup** that takes 3 parameters:

(1) a list of movie titles,
(2) a list of reviewer comments for each movie, and
(3) a movie title.

You may assume the two lists have the same length and that the 1st entry in the comment list corresponds to the 1st movie in the titles list, etc.

The function should search the titles list for the index of the title passed as the third parameter and return the corresponding comment at that index.

Return 'no comment' if the title parameter is not in the title list.

SAVE YOUR NOTEBOOK AND SUBMIT TO CANVAS:

- lab_final.ipynb
- lab_final.pdf