PPDS Exam

version 06 - Jan 14, 2023

1 Question Random and Modules + loops and conditionals (20)

How to Play PIG (background)

The object of the game Pig is to be the first player to earn 100 points. You achieve this by rolling the dice and adding which number you roll to your overall score. Players are permitted to roll as often as they'd like during their turn, but beware of rolling a 1! Doing so will cost you all the points you've collected during your turn.

If you've racked up the points and don't want to risk rolling a 1, you may choose to hold and keep the numbers you've accumulated during your turn. It will be up to you to decide if the risk is worth it! Once a player reaches 100 points, the game ends, and that player is the winner.

In this task, we will create a python program that simulates tossing two 6-sided dice to determine a winner between 2 players. You can imagine this as a part of a digital game where players can cheat to increase their chance of winning.

a. remember to `import random`

b. Create a function called `roll_dice(times, cheat)` that rolls the dice that takes 2 parameters, the number of rolls, and if the player cheats (0 or 1 for cheater). Be sure to place this and the following functions in `practice_01.py`

c. In the function `roll_dice()` create the following scenario:

- Use the random function to create an integer between 1 and 6.
- Define `score` variable to count the scores for the number rolls defined by `times`
- Use a loop function to roll the dice and add the score.
- Have a conditional in the loop function to add 5 points to the score for `cheat`
- Return the score like this `return 'score is', score`

d. From the `test01.py` call the roll_dice like this `(roll_dice(15,1),(roll_dice(15)` remember modules and to print the result.

2 Question List of Lists / For Loops (25)

In this task, we need to transpose (switch the rows and the columns if thinking about a table) the `top_hits`. The transposition of a list of lists is a new list of lists where the rows of the original become the columns, and the columns become the rows.

Once you have transposed the matrix, we create a new matrix with the `Peak`, and `Song` columns.

a. Start with creating and copying the `top_hits` in the `practice01.py` (see below)

b. In `test01.py` create a function called `transpose_list()` that uses a for loop (comprehension not allowed) to transpose items and the subitems from a "39x4" matrix to a "39x4" matrix.

- Name the new list `new_list`
- Remember to make a nested loop for the items and subitems. Use the following names:
    - Create a `for loop` inside another `for loop`
    - For the first loop, the `item` is the variable.
    - For the second loop, use `for subitem in...`
- From `test01.py,` print out the first and sixth items of the `new_list` list. (RUNNING UP THAT HILL)

c. Create a second function, `get_peak()`, that takes the `new_list` list and extracts `song` and `peak`.

- Name this new list `peak_list=[]`
- Create the loop with `for x in...` for the extraction
- Use a conditional statement that tests if `Peak` is an integer of `1` if true then insert into a new item "Number One Ht".
- Sort the `peak_list`

d. From `test01.py` run the `get_peak()` and print `(peak_list[2][1],practice01.peak_list[2][2])`.

```
top_hits=[['Pos', 'Song', 'Artist', 'Peak'],['1', 'AS IT WAS', 'HARRY STYLES', '1'],
['2', 'BAD HABITS', 'ED SHEERAN', '1'], ['3', 'PERU', 'FIREBOY DML & ED SHEERAN', '2'],
          ['4', 'GO', 'CAT BURNS', '2'], ['5', 'SHIVERS', 'ED SHEERAN', '1'], ['6',
'RUNNING UP THAT HILL', 'KATE BUSH', '1'],
          ['7', 'HEAT WAVES', 'GLASS ANIMALS', '5'], ['8', 'WHERE ARE YOU NOW', 'LOST
FREQUENCIES/CALUM SCOTT', '3'], ['9', 'AFRAID TO FEEL', 'LF SYSTEM', '1'],
          ['10', 'SEVENTEEN GOING UNDER', 'SAM FENDER', '3'], ['11', "WE DON'T TALK
ABOUT BRUNO", 'GAITAN/CASTILLO/ADASSA/FELIZ', '1'], ['12', 'MAKE ME FEEL GOOD',
'BELTERS ONLY FT JAZZY', '4'],
          ['13', 'COLD HEART', 'ELTON JOHN & DUA LIPA', '1'], ['14', 'STARLIGHT',
'DAVE', '1'], ['15', 'GREEN GREEN GRASS', 'GEORGE EZRA', '3'],
          ['16', 'WHERE DID YOU GO', 'JAX JONES FT MNEK', '7'], ['17', 'ABCDEFU',
'GAYLE', '1'], ['18', 'BABY', 'AITCH/ASHANTI', '2'],
          ['19', 'ABOUT DAMN TIME', 'LIZZO', '3'], ['20', "I AIN'T WORRIED",
'ONEREPUBLIC', '3'],
          ['21', 'CRAZY WHAT LOVE CAN DO', 'DAVID GUETTA/HILL/HENDERSON', '5'], ['22',
'EASY ON ME', 'ADELE', '1'], ['23', 'DOWN UNDER', 'LUUDE FT COLIN HAY', '5'],
          ['24', 'B.O.T.A. (BADDEST OF THEM ALL)', 'ELIZA ROSE/INTERPLANETARY', '1'],
          ['25', "I'M GOOD (BLUE)", 'DAVID GUETTA & BEBE REXHA', '1'], ['26', 'LAST
LAST', 'BURNA BOY', '4'], ['27', 'ANOTHER LOVE', 'TOM ODELL', '10']]
```

3 Question Numpy + Plots (25 points:)

In this task, we will work with a dataset of fast-food restaurant list of burgers that contains `item`, `Cateorgy`, `height`, `Calories`, `Fat(g)`, `SatFat(g)`, `TransFat(g)`, `Cholesterol(mg)`, `Sodium(mg)`, `TotalCarb(g)`, and `DietaryFiber(g)`, `Sugars(g)`, and `Protein(g)`. We will use NumPy and Matplotlib to investigate the relationships between them.

a. Use `import numpy as np` and `import matplotlib.pyplot as plt` in the `test01.py`.

- In `practice01.py` load the health dataset into a NumPy array using the `genfromtxt` function with the `skip_header` parameter enabled.
- Print the `shape` of the dataset (number of rows and columns) from `test01.py` remember to import matplotlib

b. Extract columns into variables for `calories`, `fat`, `cholesterol`, `sodium`, and `protien` columns.

c. Create a histogram from the `calories` in `test01.py`

- X-axis label "Calories"
- Title "Calories of Burgers"

d. Create a scatter plot in `test01.py` of fat vs protein, with calories level represented by color.

- X-axis label "Fat"
- Y-axis label "Protein"
- Title "Fat vs. Protein"
- Colorbar "Calories"