Project 1 - Pytzee

Due Date: Friday, November 1st, 2024 by 11:59:59 PM
Value: 90 points

This assignment falls under the standard cmsc201 academic integrity policy. This means you should not discuss/show/copy/distribute your solutions, your code or your main ideas for the solutions to any other student. Also, you should not post these problems on any forum, internet solutions website, etc.

```
"""
File:    FILENAME.py
Author:  YOUR NAME
Date:    THE DATE
Section: YOUR DISCUSSION SECTION NUMBER
E-mail:  YOUR_EMAIL@umbc.edu
Description:
  DESCRIPTION OF WHAT THE PROGRAM DOES
"""
```

UMBC

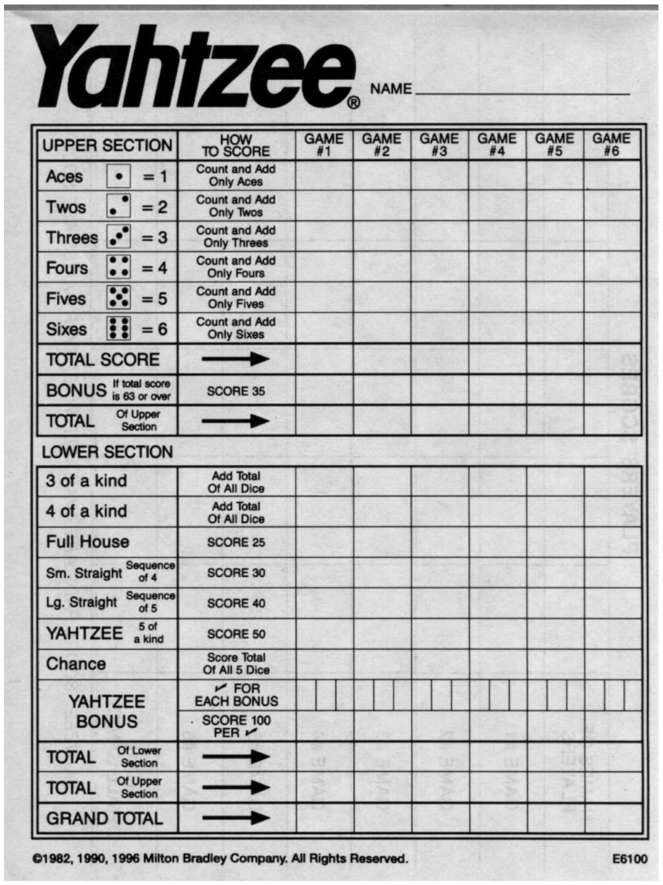
Frequently Asked Questions

- For large and small straights, the numbers do not have to be in order to count. For example, a roll of 3, 1, 2, 4, 5 is still considered a large straight.
- For three and four of a kind, you can have more than the required amount (aka 3, 3, 3, 3, 5 can be put into three of a kind)
- The 35 point bonus is applied when the count # scores are added up and equal 63 or more points. This bonus is only added once per game.

Project General Description

Yahtzee is a dice game where you try to maximize your score based on filling slots with various scores based on your rolls. Whoever has the highest score will win.

For our version we'll just do a single player version. We'll try to reproduce as much of the functionality of the game as possible but also simplifying it to make it more of a project 1 for this class.



Yahtzee® NAME ____________________

UPPER SECTION	HOW TO SCORE	GAME #1	GAME #2	GAME #3	GAME #4	GAME #5	GAME #6
Aces ⚀ = 1	Count and Add Only Aces						
Twos ⚁ = 2	Count and Add Only Twos						
Threes ⚂ = 3	Count and Add Only Threes						
Fours ⚃ = 4	Count and Add Only Fours						
Fives ⚄ = 5	Count and Add Only Fives						
Sixes ⚅ = 6	Count and Add Only Sixes						
TOTAL SCORE	→						
BONUS If total score is 63 or over	SCORE 35						
TOTAL Of Upper Section	→						
LOWER SECTION							
3 of a kind	Add Total Of All Dice						
4 of a kind	Add Total Of All Dice						
Full House	SCORE 25						
Sm. Straight Sequence of 4	SCORE 30						
Lg. Straight Sequence of 5	SCORE 40						
YAHTZEE 5 of a kind	SCORE 50						
Chance	Score Total Of All 5 Dice						
YAHTZEE BONUS	✓ FOR EACH BONUS						
	SCORE 100 PER ✓						
TOTAL Of Lower Section	→						
TOTAL Of Upper Section	→						
GRAND TOTAL	→						

©1982, 1990, 1996 Milton Bradley Company. All Rights Reserved. E6100

Here's a scorecard from the game, where each person fills in their score on their card.

Rules of the Game

Here is the outline of the rules of our game, keep in mind that we're modifying the rules for the game, so don't just look up the rules for the original, use these.

1. Ask the user for the number of rolls they want to make. This determines the number of turns in the game.
2. When a roll is made, score in this way:
    a. If the user selects to count 1's, 2's, 3's, 4's, 5's, or 6's, then count the number of dice with that number, and then give the user the number of points equal to the value of the die times the number of times it occurs.
        i. For instance 3 4's will give 12 points, 5 1's will give 5 points, 5 6's will give 30 points.
        ii. Only allow the user to choose each value once, so once the user enters a number for 4's, they cannot do so again, even if they get more 4's in the next rolls.
    b. If the user selects to count Three of a kind or four of a kind, then give them the sum of all the dice that they rolled, including those which are not part of the 3 or 4 of a kind. For instance if the four of a kind is 2, 2, 2, 2, 6, then you will award 2 + 2 + 2 + 2 + 6 points.
    c. A full house is a three of a kind and a pair of different types, if this happens and the user selects to use the full house rather than the three of a kind, then give them 25 points.
    d. If the user gets a small straight, which is 4 in a row, either 1, 2, 3, 4 or 2, 3, 4, 5, or 3, 4, 5, 6, then award 30 points.

e. If the user gets a large straight, meaning five in sequence so either 1, 2, 3, 4, 5 or 2, 3, 4, 5, 6, then award 40 points.
f. If the user selects pytzee and has all five dice the same, then they get a pytzee and 50 points.
    i. If they get another pytzee, they can earn 100 points per additional pytzee.
g. If the user selects chance, then add the sum of all the dice rolls and count that in the chance box. It doesn't have to be any configuration for instance 2, 4, 5, 6, 6 is not really anything in particular but will count as 2 + 4 + 5 + 6 + 6 points in a chance box.

3. To sum the user's score, add all the 1's, 2's, 3's, 4's, 5's, and 6's scores and then if that score is at least 63, add a bonus of 35. (This bonus is only added once).
4. Unlike regular Yahtzee and in order to simplify, there aren't any re-rolling of specific dice, the user gets whatever their first roll is.
5. There must also be an option for a user to "skip" since for instance if they've already used their three of a kind, chance, and 1's and get 1, 1, 1, 2, 3, then there isn't anything to record the score in, so the dice roll is ignored and the turn ends.

Design Overview

Your project should include at least the functions:

1.) This function should be the entry point for your program. Your if __name__ == "__main__": can be composed of the following:
    a.) Input the number of rounds as an integer.
    b.) Input the seed as an integer.
    c.) Set the seed.
    d.) Call play_game.
2.) The function to play the game should be called:

```
def play_game(num_rounds):
```

    a.) Do not change the parameter of this function or its name.
    b.) We will test your code on specific maps by calling this function, so if you rename it, the tests will not run correctly.
3.) You should create at least 4 additional functions for your project. Suggestions include:
    a.) A function to get the number of each type of dice rolled.
    b.) Functions to detect whether you have a three of a kind, four of a kind, pytzee, small or large straight, etc.
    c.) Functions to handle scoring.
    d.) You can make other functions than these as well.
4.) Your game should do the following:
    a.) Display the turn and current score.
    b.) Roll the dice and display the results.
    c.) Ask the user what they want to count the dice roll as, or skip.
        i.) Repeat until they enter a valid possibility and either the dice roll is counted for points or is skipped.
        ii.) You should accept "3 of a kind" or "three of a kind", "4 of a kind" or "four of a kind", “full house”, "chance", "pytzee", "small straight" and "large straight".

iii.) If they want to count the 1's, 2's, 3's, 4's, 5's, or 6's, accept the input "count 1", "count 2", "count 3". You **don't** have to accept "count 1s" since that would require more string parsing to get the 1 separate from the s before you cast.

iv.) I generally recommend using .lower() on the input but we'll only use lowercase for testing purposes to avoid problems.

d.) Display the player's current score card.

i.) In order to do this, and save space on the screen I used tabs in print statements so for instance to print out the scorecard I used end='\t' to space out the results.

ii.) First I print out the headers, then I print out the scores you can't go back up in the line, just keep that in mind.

iii.) For the Three of a Kind, Four of a Kind, I used the following code `str(x).rjust(len(SCORE_TYPES[i])).` SCORE_TYPES is a list containing the names of each of the scoring types, i.e. "Three of a Kind", and x represents the score for that value. You are free to display the score however you want, horizontally, vertically, etc. This is just how my code does it.

iv.) Display 0 for any unfilled box.

e.) In order to reduce the vast number of string constants that this may create, any input that we check from the user, so for instance if you have a statement like

if command == "full house", you don't have to use a constant like FULL_HOUSE = 'full house' though if you do it's probably slightly better.

f.) You are **permitted** to use dictionaries but this project can be done entirely with one dimensional lists.

Provided Functions

I have provided a starter file on the GL server at:

/afs/umbc.edu/users/e/r/eric8/pub/cs201/spring22/pytzee_starter.py

You should copy it to your directory using the command:

```
cp /afs/umbc.edu/users/e/r/eric8/pub/cs201/spring22/pytzee_starter.py pytzee.py
```

```
roll_dice()
```

This function can be called as roll_dice(). It will roll five six-sided dice, and return the result as a list of integers. You should not modify this function and should only call it once per turn because calling it extra times will result in messing up the random sequence.

Make sure to copy using the command and don't submit pytzee_starter.py. If you do, points will be deducted. I am using a different file name so that accidentally copying it won't eliminate any code you have already in pytzee.py.

Submission Details

Submit the files under the following titles:
(These are case sensitive as usual.)
submit cmsc201 PROJECT1 pytzee.py

Coding Standards

Coding standards can be found here.

1. At least one inline comment per function explaining something about your code.
2. Constants above your function definitions, outside of the "if __name__ == '__main__':" block.
    a. A magic value is a string which is outside of a print or input statement, but is used to check a variable, so for instance:
        i. `print(first_creature_name, 'has died in the fight. ')` does not involve magic values.
        ii. However, `if my_string == 'EXIT':` exit is a magic value since it's being used to compare against variables within your code, so it should be:
        ```
        EXIT_STRING = 'EXIT'
        if my_string == EXIT_STRING:
        ```
    b. A number is a magic value when it is not 0, 1, and if it is not 2 being used to test parity (even/odd).
    c. A number is magic if it is a position in an array, like my_array[23], where we know that at the 23rd position, there is some special data. Instead it should be
    USERNAME_INDEX = 23
    my_array[USERNAME_INDEX]

    d. Constants in mathematical formulas can either be made into official constants or kept in a formula.

3. Previously checked coding standards involving:
    a. snake_case_variable_names
    b. CAPITAL_SNAKE_CASE_CONSTANT_NAMES
    c. Use of whitespace (2 before and after a function, 1 for readability.)

Allowed Built-ins/Methods/etc

- Declaring and assigning variables, ints, floats, bools, strings, lists, dicts.
- Using +, -, *, /, //, %, **; +=, -=, *=, /=, //=, %=, **= where appropriate
- Comparisons ==, <=, >=, >, <, !=, in
- Logical and, or, not
- if/elif/else, nested if statements
- Casting int(x), str(x), float(x), (technically bool(x))
- For loops, both *for i* and *for each* type.
- While loops
    - sentinel values, boolean flags to terminate while loops
- Lists, list(), indexing, i.e. my_list[i] or my_list[3]
    - 2d-lists if you want them/need them my_2d[i][j]
    - Append, remove, del
    - **list slicing**
- If you have read this section, then you know the secret word is: createous.
- String operations, concatenation +, +=, split(), strip(), join(), upper(), lower(), isupper(), islower()
    - **string slicing**
- Print, with string formatting, with end= or sep=:
    - '{}'.format(var), '%d' % some_int, f-strings
    - Really the point is that we don't care how you format strings in Python
    - Ord, chr, but you won't need them this time.
- Input, again with string formatting in the prompt, casting the returned value.
- Using the functions provided to you in the starter code.
- Using import with libraries and specific functions **as allowed** by the project/homework.

- You may define your own new functions with new parameters
    - Single return values (you must return a single float, bool, string, int, list or dictionary, or None/not return).
- **For this project, the sum keyword is permitted.**
- **For this project, dictionaries are permitted but not necessary.**

Forbidden Built-ins/Methods/etc

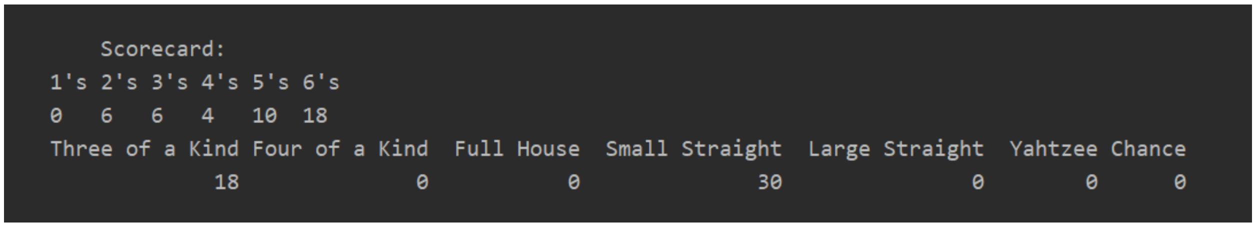
This is not a complete listing, but it includes:

- Multiple returns [i.e. `var1, var2 = function_call()`]
- break, continue
- methods outside those permitted within allowed types
    - for instance str.endswith
    - list.index, list.count, etc.
- Keywords you definitely don't need: await, as, assert, async, class, except, finally, global, lambda, nonlocal, raise, try, yield
- The *is* keyword is forbidden, not because it's necessarily bad, but because it doesn't behave as you might expect (it's not the same as ==).
- built in functions: any, all, breakpoint, callable, classmethod, compile, exec, delattr, divmod, enumerate, filter, map, max, min, isinstance, issubclass, iter, locals, oct, next, memoryview, property, repr, reversed, set, setattr, sorted, staticmethod, super, type, vars, zip.
- If you have read this section, then you know the secret word is: serendipity.
- The two built in functions exit() and quit()
- If something is not on the allowed list, not on this list, then it is probably forbidden.
- The forbidden list can always be overridden by a particular problem, so if a problem allows something on this list, then it is allowed for that problem.
- Note that we are using python 3.9.7 currently on the GL server so new features like switch/case are 'forbidden' but only in the sense that it doesn't exist in our current version.

Sample Project Output

Here is a single quick run of the program, for more you should run the pyc file. I know it's a bit hard to read the output from the 3, 4, Full House... part of the card. Look at the pyc output to see it better.

The output actually looks like this:



```
    Scorecard:
1's 2's 3's 4's 5's 6's
0   6   6   4   10  18
Three of a Kind Four of a Kind  Full House  Small Straight  Large Straight  Yahtzee Chance
             18               0           0              30               0       0      0
```

```
linux3[25]% python3 jmps_and_hlts.py
What is the number of rounds that you want to play? 10
Enter the seed or 0 to skip: 17
***** Beginning Round 1 *****
    Your score is: 0
    5    4    3    3    3
How would you like to count this dice roll? three of a kind
Three of a Kind!

     Scorecard:
    1's    2's    3's    4's    5's    6's
    0    0    0    0    0    0
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
              18                  0             0
 0                  0         0         0

***** Beginning Round 2 *****
    Your score is: 18
    2    6    6    5    6
```

```
How would you like to count this dice roll? count 6
Accepted the 6

     Scorecard:
    1's    2's    3's    4's    5's    6's
    0    0    0    0    0    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
              18                  0             0
     0                  0         0         0

***** Beginning Round 3 *****
    Your score is: 36
    3    1    1    2    4
How would you like to count this dice roll? small straight
You have a small straight and get 30 points.

     Scorecard:
    1's    2's    3's    4's    5's    6's
    0    0    0    0    0    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
              18                  0             0             30
     0        0         0

***** Beginning Round 4 *****
    Your score is: 66
    6    4    3    5    3
How would you like to count this dice roll? count 3
Accepted the 3

     Scorecard:
    1's    2's    3's    4's    5's    6's
    0    0    6    0    0    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
              18                  0             0             30
     0        0         0
```

```
***** Beginning Round 5 *****
    Your score is: 72
    6    6    6    4    2
How would you like to count this dice roll? count 6
There was already a score in that slot.
How would you like to count this dice roll? skip

      Scorecard:
    1's     2's     3's     4's     5's     6's
    0    0    6    0    0    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
             18                  0              0             30
     0         0         0

***** Beginning Round 6 *****
    Your score is: 72
    5    1    2    2    2
How would you like to count this dice roll? count 2
Accepted the 2

      Scorecard:
    1's     2's     3's     4's     5's     6's
    0    6    6    0    0    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
             18                  0              0             30
     0         0         0

***** Beginning Round 7 *****
    Your score is: 78
    6    5    5    6    2
How would you like to count this dice roll? count 5
Accepted the 5

      Scorecard:
    1's     2's     3's     4's     5's     6's
    0    6    6    0    10    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
             18                  0             0             30
 0         0         0

***** Beginning Round 8 *****
    Your score is: 88
    3    5    1    6    6
How would you like to count this dice roll? skip

      Scorecard:
    1's    2's    3's    4's    5's    6's
    0    6    6    0    10    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
             18                  0             0             30
 0         0         0

***** Beginning Round 9 *****
    Your score is: 88
    1    3    4    1    5
How would you like to count this dice roll? skip

      Scorecard:
    1's    2's    3's    4's    5's    6's
    0    6    6    0    10    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
             18                  0             0             30
 0         0         0

***** Beginning Round 10 *****
    Your score is: 88
    4    6    5    2    6
How would you like to count this dice roll? count 2
There was already a score in that slot.
How would you like to count this dice roll? count 4
Accepted the 4
```

```
     Scorecard:
    1's    2's    3's    4's    5's    6's
    0    6    6    4    10    18
    Three of a Kind    Four of a Kind    Full House    Small
Straight    Large Straight    Yahtzee    Chance
               18                0              0             30
     0         0         0

Your final score was 92
```

UMBC

Sample Runs - PYC File

You can run my version of the project and see some sample output.

I'm going to release a compiled version of the project so that you can play the game and get a feel for what the sample output should be. You don't have to match my messages exactly but the program should flow in the same way.

The pyc file (a compiled python file) can be found at /afs/umbc.edu/users/e/r/eric8/pub/cs201/fall24/pytzee.pyc

To copy a file you should use cp and then . at the end. Without the period at the end it won't copy to the current directory.

Run the file by running the command:

```
linux5[115]% cp /afs/umbc.edu/users/e/r/eric8/pub/cs201/fall24/pytzee.pyc .
linux5[116]% python3 pytzee.pyc
```

It may only work on the GL server or on linux with python3 version 3.9.7

Approximate Rubric

I say approximate here to mean that this rubric doesn't break down specific requirements within each group, and may be modified by a few points here and there, but this should provide you with a good understanding of what we intend the value of each piece of the project implementation to be.

Requirement	**Points** (appx)
Loop the required number of turns.	10
Display the score and the score card to the user	20
Ask the user what to do with their dice rolls, and handle the dice rolls.	20
Handle Bonus scores for extra pytzees and score counts above 63 for the dice counts.	10
Detect pytzees, four of a kinds, three of a kinds, full houses, small and large straights.	20
Coding Standards, Documentation, Comments, Overall Correctness	10