Cee-lo game

Cee-lo is a gambling game played with three six-sided dice. In this paper, I will implement a program in Python which plays the variant "**Cee-lo with a bank**".

How is the game played?

The game is played with two or more players. At the beginning of the game each player has the same number of coins (or chips). In each round, one player is designated as the banker (also known as the "mana") and all other players bet against him. If a player, for example, bets 2 coins, he will give the banker or take 2 coins from him, depending on his roll and the banker's roll. The banker has a slight advantage over other players in terms of the chances of winning a bet. The role of banker is passed from one player to another throughout the game. At the beginning of the round, the banker places an amount B in the center. The other players, in turn, starting with the player who is next to the banker (e.g. the player sitting to the banker's left) bet an amount, which cannot exceed B minus the total amount bet by the previous players. For example, if B = 5 and the first player bets 3, the next player cannot bet more than 2. If the entire amount is covered by the bets of the first players, those who have not yet bet do not participate in the round. If the full amount is not covered after all players have bet, then the uncovered amount is returned to the banker, so the total amount of the "bank" may be less than the initial B. After all bets are finalized, the banker rolls the dice. There are four possible outcomes from his roll: automatic win, automatic loss, score setting, or reroll.

- **Automatic win**: if the banker rolls 4-5-6 or "triple" (all three dice show the same number) or two same numbers and a 6, then all bets are won immediately. Examples of automatic win dice: 4-5-6, 3-3-3, 2-2-6
- **Automatic loss**: if the banker rolls 1-2-3 or a pair of same numbers (other than 1) and a 1, then he immediately loses all bets (players "break" the bank). Examples of automatic losing dice: 1-2-3, 1-3-3, 1-3-3, 1-6-6
- **Score setting**: if the banker rolls two same numbers and a third number (2, 3, 4 or 5), then the third number becomes the banker's "score". For example, if he rolls 2-2-4, then his score will be 4. Note, the score cannot be 1 or 6, as these would result in an automatic loss or win, respectively (see above).
- **Reroll**: if the roll is none of the above combinations, then the banker re-rolls the dice

If the banker does not make an automatic win or loss, then he will have a score of 2, 3, 4 or 5. Then, the players who have placed bets take turns rolling the dice, starting with the player sitting after the banker, so that individual bets can be settled. A player beats the banker if the player rolls 4-5-6, "triple", or a score higher than the banker's. A player loses if he brings 1-2-3, or achieves a score lower than the banker's score. If the player bears the same score as the banker, then we have a tie and neither player wins. If the player rolls a die (e.g. 2-3-5) that does not result in a win, a loss, or a score, they re-roll the dice until a result is obtained. At the end of the round, if the banker has lost all bets with the players (this can happen if he/she brings an automatic loss or loses individually to all players), then he/she also "loses" the bank and in the next round, the banker becomes the next player at the table. Also, another case of the banker "losing" the bank in the next round is if he is

beaten by a player with 4-5-6 or "triple". The first player to do this in the current round also takes the bank in the next round. Thus, the banker keeps the bank in the next round if he (a) achieves an automatic win, (b) does not lose all bets, and (c) does not lose to a player who made a 4-5-6 or "triple" The game ends if at the end of a round a player has lost all his coins. The player with the highest number of coins is declared the winner.

Requirements

You are asked to write a program that simulates a game of "Cee-lo" as described above. Your program should ask and get from the user the number of players (between 2 and 6) and the number of coins (between 5 and 100) that each player has at the beginning of the game. For example:

```
Number of players (between 2 and 6): 4

Number of coins per player (between 5 and 100): 10
```

In the example above, the game is played between 4 players and at the beginning each player has 10 coins. In case the user gives a non-integer number of players or out of bounds, the number of players will be set to 3. In case the user gives a non-integer number of coins or out of bounds, the number of players will be set to 10:

```
Number of players (between 2 and 6): 99
I expected between 2 and 6 players
I'm setting the number of players to 3

Number of coins per player (between 5 and 100): ‘jjjjj’
Something wrong happened: invalid literal for int() with base 10: 'jjjjj'
I'm setting the number of coins to 10
```

Then, your program should randomly decide which player will be the first to play banker. For random decisions you can use the randint() function from the random library. Then, your program should simulate the game. Your program should print, at the beginning of each round, the current balance of each player. Then, in the first phase of the round, the banker determines the total amount B on which the other players will bet (B must be at least 1 and at most equal to the banker's balance). Then, players are asked to bet in turn, starting with the player who is next in line to the banker. Each player must bet at least 1 coin and no more than his/her balance and the remaining amount from B available for betting. If there is no balance available from the initial B then the player (and any others remaining) is not eligible to bet. The second phase of the round starts when all players have bet or there is no balance available to bet. For example, if 4 players are playing, the banker bets B = 5, the next player bets 3 and the next player bets 2, then there is no balance left for the last player and he does not play in the second phase. Before the second phase of the round starts, the amount bet for each player is printed and the banker's "bank" is printed, i.e. the total amount bet. In the second phase, the banker must roll the dice until he/she achieves an automatic win, automatic loss, or score determination. If he achieves an automatic win, he wins all bets, which are deducted from the players and added to the banker's balance. If he achieves an automatic loss, he loses all bets. If he scores a score, then all players roll in turn and (if they do not score 4-5-6, "triple", or 1-2-3) compare their score with the banker's score and decide who wins (or if there is a tie). After all players have played, the round is over. The program decides who will be the banker in the next round (according to the description) and whether the game is over (in case a player has a zero balance).