Project Context: The goal of your project is to create a console-based blackjack game for a video game development company. This blackjack game should automate the game's rules, allowing players to play against the computer.

In a deck of cards, each card is characterized by its rank (1 for Ace, 2 to 10 for numbered cards, 11 for Jack, 12 for Queen, 13 for King), and its suit (1 for Diamonds, 2 for Hearts, 3 for Spades, and 4 for Clubs).

Example:

[12, 3]: Queen of Spades

[5, 1]: 5 of Diamonds

You want to create a 52-card deck defined as a list of cards. To do this, you will define a method that constructs a list of cards based on a given starting card. You will then use this method to create the deck of cards starting with the Ace of Diamonds. To accomplish this, you should use:

1. A method that constructs the list of next cards from a given card as a parameter. This list should follow the order of suits (Diamonds, Hearts, Spades, Clubs), so the last card corresponds to the King of Clubs.
2. A method that creates the 52-card deck defined as a list of pairs of values.

Example: (create\_cards) -> [(1, 1), (2, 1), ... (12, 4), (13, 4)]

To shuffle the deck, you need to randomly draw a card to be the first card in the shuffled deck. To do this, you will randomly choose the index i of the card to draw and then extract the card at index i from the deck. Here are three methods that help you implement this:

1. The "extract\_ith\_card" method extracts the i-th card from a given list of cards as a parameter. This method returns a list consisting of the extracted card and the list of remaining cards.

Example: extract\_ith\_card([(1, 1), (12, 3), (14, 4)], index: 2) -> [(12, 3), [(1, 1), (14, 4)]]

1. The "draw\_a\_card" method randomly draws a card from the cards in a given list as a parameter. This function returns a list consisting of the drawn card and the list of remaining cards after the draw. To implement this method, use the "random n" function, which returns a random number between 0 (inclusive) and n (exclusive), as well as the "extract\_ith\_card" method.

Example: draw\_a\_card([(1, 2), (13, 4), (5, 3), (1, 1), (6, 3)]) -> [(6, 3), [(1, 2), (13, 4), (5, 3), (1, 1)]]

1. Write the "shuffle\_deck\_of\_cards" method, which shuffles the cards in a given list of cards as a parameter. This method returns the list of shuffled cards. Note that this method uses the "draw\_a\_card" method.

Example: shuffle\_deck\_of\_cards([(10, 1), (4, 4), (5, 2)]) -> [(10, 1), (5, 2), (4, 4)]

Drawing cards is different from randomly drawing cards: drawing cards involves a card stack. When drawing cards, you take the cards from the top of the stack. When discarding, you place the cards at the bottom of the stack.

1. The "draw\_n\_cards" method allows you to draw the first n cards from a given list of cards as a parameter. This method returns a list consisting of the drawn cards and the list of remaining cards in the stack.

Example: draw\_n\_cards([(13, 4), (5, 3), (1, 1), (6, 3)], index: 2) -> ([(13, 4), (5, 3)], [(1, 1), (6, 3)])

1. The "discard\_cards" method takes two lists of cards as parameters (a list of cards corresponding to the stack and a list of cards to discard) and returns the list of cards from the stack, with the discarded cards placed at the end.

Example: discard\_cards([(pile: [(1, 1), (1, 2)]), dealer + player cards: [(3, 4), (4, 3)]) -> [(1, 1), (1, 2), (3, 4), (4, 3)];

A game of blackjack involves the player against the dealer. The dealer starts by drawing one card and then gives two cards to the player. The goal of the game is to get as close to or hit 21 without going over it. The card values are established as follows: cards 2 to 10 retain their values, face cards are worth 10, and the Ace can be worth 1 or 11 if it helps reach 21.

The player can request as many cards as they want. If they go over 21, they lose. If they stop before that, the dealer draws cards until they exceed 17 to avoid taking risks. If the dealer ends up with less than the player or goes over 21, the player wins. If the dealer has more than the player, the player loses. Otherwise, it's a tie.

Haut du formulaire