TEST CASES

Description:

This document includes 14 test cases to validate the designed Card Game.
There are 6 functional test cases, which validate the designed game functionally working well.
There are 8 edge test cases, which validate the robustness of the designed game in case of errors and exception.

One python file named “test_case.py” with the 13 test cases is created for the user’s convenience to test the designed Card Game.

Functional Test Case:

Test Case 1:
Test the card is initiated correctly.
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

Expected Output:

```
cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

suits= ['red', 'yellow', 'green']

points= {'red': 3, 'yellow': 2, 'green': 1}

deck number= 1
```

Output:

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4)
suits= ['red', 'yellow', 'green']
points= {'red': 3, 'yellow': 2, 'green': 1}
deck number= 1

Process finished with exit code 0
```

Test Case 2:
#Test “Shuffle cards in the deck” Operation works
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("original cards:", cardgame.cards)

cardgame.shuffleCards()

print("original cards:", cardgame.cards)

print("shuffled cards", cardgame.cards)
```

Expected Output:

Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Shuffled Cards 1 =”original cards with random order”

Shuffled Cards 2 =”original cards with random order which is different from Shuffled Cards 1”

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards)

cardgame.shuffleCards()

print("Shuffled Cards 1 =", cardgame.cards)

cardgame.shuffleCards()

print("Shuffled Cards 2 =", cardgame.cards)

```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]
Shuffled Cards 1 = [('green', 0), ('green', 5), ('red', 1), ('yellow', 0), ('yellow', 2), ('green', 4), ('green', 3), ('yellow', 1), ('red', 0), ('green', 1), ('green', 2), ('yellow', 3)]
Shuffled Cards 2 = [('green', 5), ('red', 1), ('green', 3), ('green', 0), ('red', 0), ('yellow', 3), ('green', 1), ('yellow', 2), ('yellow', 1), ('yellow', 0), ('green', 2), ('green', 4)]
```

Test Case 3:
Test “Get a card from the top of the deck” Operation works

Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("Original Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')
```

Expected Output:

```
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]
Card Number= 12

Deal Card= ('green', 5)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4)]
Card Number= 11

Deal Card= ('green', 4)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3)]
```

Card Number= 10

Deal Card= ('green', 3)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2)]
Card Number= 9

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

print("Cards=", cardgame.cards)

print("Card Number=", len(cardgame.cards), '\n')

```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]
Card Number= 12

Deal Card= ('green', 5)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4)]
Card Number= 11

Deal Card= ('green', 4)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3)]
Card Number= 10

Deal Card= ('green', 3)
Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2)]
Card Number= 9


Process finished with exit code 0
```

Test Case 4:
Test “Sort cards” Operation Works
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],
                    points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.shuffleCards(seed=1)
```

print("Shuffled Cards=", cardgame.cards, '\n')

cardgame.sortCards(suits_order=["yellow", "green", "red"])

print("Sorted Cards=", cardgame.cards)

Expected Output:

Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Shuffled Cards= [('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]

Sorted Cards= [('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5), ('red', 0), ('red', 1)]

Output:

```
1  from CardGame import *
2
3  cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],
4                      points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)
5
6  print("Original Cards=", cardgame.cards, '\n')
7
8  cardgame.shuffleCards(seed=1)
9
10 print("Shuffled Cards=", cardgame.cards, '\n')
11
12 cardgame.sortCards(suits_order=["yellow", "green", "red"])
13
14 print("Sorted Cards=", cardgame.cards)
15
```

```
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Shuffled Cards= [('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]

Sorted Cards= [('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5), ('red', 0), ('red', 1)]

Process finished with exit code 0
```

Test Case 5:

Test “Determine winners” Operation works

Input:

from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'], points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.play()

Expected Output:

Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Card Suit - Point: {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards: [('green', 5), ('green', 3), ('green', 1)]
Player 2's cards: [('green', 4), ('green', 2), ('green', 0)]

Player 1 Wins!(Points: 9)

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards)

cardgame.play()

```

Test Case 6:
Test “Determine winners” Operation works with reproducible Shuffling
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.shuffleCards(seed=1)

print("Shuffled Cards=", cardgame.cards, '\n')

print(cardgame.cards)

cardgame.play()

cardgame.play()
```

Expected Output:

Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Shuffled Cards= [('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]

[('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]
Card Suit - Point: {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards: [('yellow', 0), ('red', 1), ('green', 4)]
Player 2's cards: [('green', 3), ('yellow', 2), ('yellow', 1)]

Player 2 Wins!(Points: 9)
Card Suit - Point: {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards: [('green', 0), ('green', 2), ('green', 5)]
Player 2's cards: [('yellow', 3), ('red', 0), ('green', 1)]

Tie (Points: 7, 7)

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.shuffleCards(seed=1)

print("Shuffled Cards=", cardgame.cards, '\n')

print(cardgame.cards)

cardgame.play()

cardgame.play()
```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

Shuffled Cards= [('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]

[('green', 1), ('green', 5), ('red', 0), ('green', 2), ('yellow', 3), ('green', 0), ('yellow', 1), ('green', 4), ('yellow', 2), ('red', 1), ('green', 3), ('yellow', 0)]
Card Suit - Point:  {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards:  [('yellow', 0), ('red', 1), ('green', 4)]
Player 2's cards:  [('green', 3), ('yellow', 2), ('yellow', 1)]

Player 2 Wins!(Points: 9)
Card Suit - Point:  {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards:  [('green', 0), ('green', 2), ('green', 5)]
Player 2's cards:  [('yellow', 3), ('red', 0), ('green', 1)]

Tie (Points: 7, 7)

Process finished with exit code 0
```

Edge Test Case:

Test Case 7:
Test the case of more than one deck
Input:

from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'], points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=2)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)

Expected Output:

cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5), ('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

suits= ['red', 'yellow', 'green']

points= {'red': 3, 'yellow': 2, 'green': 1}

deck number= 2

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=2)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)

```

```
Run:  test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
cards= [('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5),

('red', 0), ('red', 1), ('yellow', 0), ('yellow', 1), ('yellow', 2), ('yellow', 3), ('green', 0), ('green', 1), ('green', 2), ('green', 3), ('green', 4), ('green', 5)]

suits= ['red', 'yellow', 'green']

points= {'red': 3, 'yellow': 2, 'green': 1}

deck number= 2

Process finished with exit code 0
```

Test Case 8:
Test the case where the number of a suit is less than 1
Input:

from CardGame import *

cardgame = CardGame(card_number=[2, 0, 6], suits=['red', 'yellow', 'green'], points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)

Expected Output:

ValueError: The card number of any suit should be greater than 0.

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 0, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=2)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Traceback (most recent call last):
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py", line 4, in <module>
    deck_number=2)
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/CardGame.py", line 26, in __init__
    self.cards.extend(Card(card_number=card_number, suits=suits).cards)
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/CardGame.py", line 8, in __init__
    raise ValueError("The card number of any suit should be greater than 0.")
ValueError: The card number of any suit should be greater than 0.

Process finished with exit code 1
```

Test Case 9:
Test the case where the card number for suits does not match suits
Input:

from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6, 3], suits=['red', 'yellow', 'green'], points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)

Expected Output:

ValueError: The card number and suit number are not matching.

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6, 3], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=2)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Traceback (most recent call last):
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py", line 4, in <module>
    deck_number=2)
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/CardGame.py", line 19, in __init__
    raise ValueError("The card number and suit number are not matching.")
ValueError: The card number and suit number are not matching.

Process finished with exit code 1
```

Test Case 10:
Test case where user-assigned suits does not match user-assigned suit-point pairs (number of items do not match)
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['red', 'yellow', 'green'],  points={'red': 3, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

Expected Output:

ValueError: The input suits and input points are not matching.

Output:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6, 3], suits=['red', 'yellow', 'green'],  points={'red': 3, 'green': 1},
                    deck_number=2)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

Test Case 11:
Test case where user assigned suits do not match user assigned suit-point pairs (terms of items do not match)
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 4, 6], suits=['blue', 'yellow', 'green'],  points={'red': 3, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

print("suits=", cardgame.suits, '\n')

print("points=", cardgame.points, '\n')

print("deck number=", cardgame.deck_number)
```

Expected Output:

ValueError: The input suits and input points are not matching.

Output:

Test Case 12:
Test case where there is no card left to deal
Input:

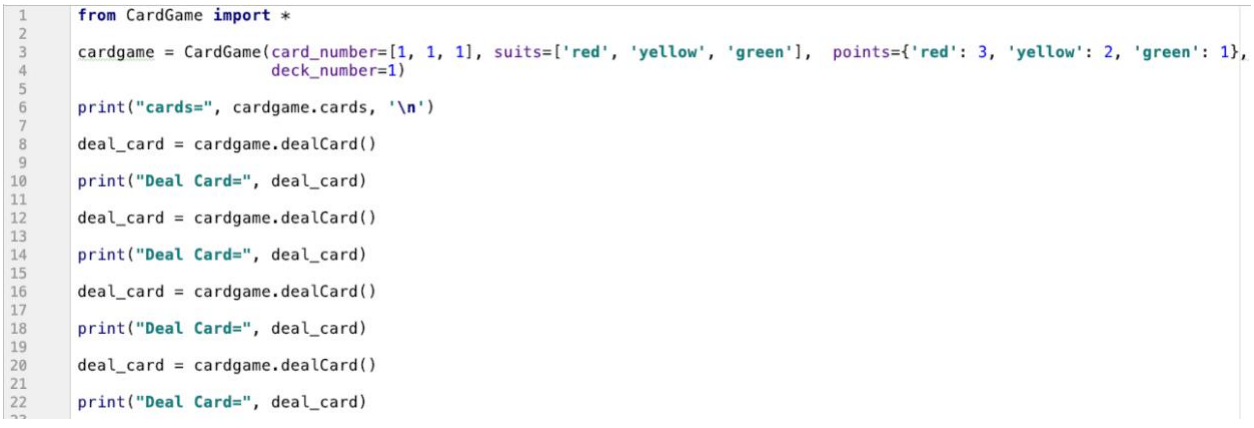
```
from CardGame import *
```

```
cardgame = CardGame(card_number=[1, 1, 1], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1}, deck_number=1)

print("cards=", cardgame.cards, '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)
```

Expected Output:

```
Deal Card= ('green', 0)
Deal Card= ('yellow', 0)
Deal Card= ('red', 0)
ValueError: There is no card left in the deck.
```

Output:



```
from CardGame import *

cardgame = CardGame(card_number=[1, 1, 1], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("cards=", cardgame.cards, '\n')

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)

deal_card = cardgame.dealCard()

print("Deal Card=", deal_card)
```

Test Case 13:
Test the case where there are duplicates in user-assigned suits
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[1, 1, 1], suits=['red', 'red', 'green'],  points={'red': 3, 'green': 1}, deck_number=1)

print('cards=', cardgame.cards, "\n")
```

Expected Output:

ValueError: The suits should have unique suit.

Output:

Test Case 14:
Test case where the left cards are not enough to play to get a winner
Input:

```
from CardGame import *

cardgame = CardGame(card_number=[2, 1, 3], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.shuffleCards(seed=1)
```

print("Shuffled Cards=", cardgame.cards, '\n')

print(cardgame.cards)

cardgame.play()

cardgame.play()

Expected Output:

Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('green', 0), ('green', 1), ('green', 2)]

Shuffled Cards= [('yellow', 0), ('green', 0), ('green', 2), ('red', 0), ('green', 1), ('red', 1)]

[('yellow', 0), ('green', 0), ('green', 2), ('red', 0), ('green', 1), ('red', 1)]
Card Suit - Point: {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards: [('red', 1), ('red', 0), ('green', 0)]
Player 2's cards: [('green', 1), ('green', 2), ('yellow', 0)]

ValueError: The number of cards left is not enough to play the game.

Output:

```python
from CardGame import *

cardgame = CardGame(card_number=[2, 1, 3], suits=['red', 'yellow', 'green'],  points={'red': 3, 'yellow': 2, 'green': 1},
                    deck_number=1)

print("Original Cards=", cardgame.cards, '\n')

cardgame.shuffleCards(seed=1)

print("Shuffled Cards=", cardgame.cards, '\n')

print(cardgame.cards)

cardgame.play()

cardgame.play()
```

```
test_case
"/Users/kangjian_ma/.conda/envs/Essex Management Code Interview/bin/python" "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py"
Original Cards= [('red', 0), ('red', 1), ('yellow', 0), ('green', 0), ('green', 1), ('green', 2)]

Shuffled Cards= [('yellow', 0), ('green', 0), ('green', 2), ('red', 0), ('green', 1), ('red', 1)]

[('yellow', 0), ('green', 0), ('green', 2), ('red', 0), ('green', 1), ('red', 1)]
Card Suit - Point:  {'red': 3, 'yellow': 2, 'green': 1}

Player 1's cards:  [('red', 1), ('red', 0), ('green', 0)]
Player 2's cards:  [('green', 1), ('green', 2), ('yellow', 0)]

Tie (Points: 3, 3)
Traceback (most recent call last):
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/test_case.py", line 16, in <module>
    cardgame.play()
  File "/Users/kangjian_ma/Desktop/Essex Management Code Interview/CardGame.py", line 61, in play
    raise ValueError("The number of cards left is not enough to play the game.")
ValueError: The number of cards left is not enough to play the game.

Process finished with exit code 1
```