```
class CardTests(unittest.TestCase):
    ## Test that if you create a card with rank 12, its rank will be "Queen"
    def test_cardrank(self):
        c = Card(rank=12)
            self.assertEqual(c.rank,"Queen")

        ## Test that if you create a card with rank 1, its rank will be "Ace"
        def test_cardrank1(self):
            c = Card(rank=1)
                self.assertEqual(c.rank,"Ace")

        ## Test that if you create a card instance with rank 3, its rank will
            be 3
        def test_cardrank2(self):
            c = Card(rank=3)
                self.assertEqual(c.rank,3)


        ## Test that if you create a card instance with suit 1, it will be suit
            "Clubs"
        def test_cardsuit(self):
            c = Card(suit=1)
                self.assertEqual(c.suit,"Clubs")

        ## Test that if you create a card instance with suit 2, it will be suit
            "Hearts"
        def test_cardsuit1(self):
            c = Card(suit=2)
                self.assertEqual(c.suit,"Hearts")


## Test that if you create a card instance, it will have access to a variable
    suit_names that contains the list ["Diamonds","Clubs","Hearts","Spades"]
def test_cardInstance(self):
    c = Card()
        self.assertEqual(c.suit_names,["Diamonds","Clubs","Hearts","Spades"])

        ## Test that if you invoke the __str__ method of a card instance that
            is created with suit=2, rank=7, it returns the string "7 of Hearts"
        def test_cardInstance2(self):
            c = Card(suit=2, rank = 7)
                self.assertEqual(str(c),"7 of Hearts")

        ## Test that if you create a deck instance, it will have 52 cards in
            its cards instance variable
        def test_deckInstance(self):
            d = Deck()
                self.assertEqual(len(d.cards),52)

        ## Test that if you invoke the pop_card method on a deck, it will
            return a card instance.
        def test_popInstance(self):
            d = Deck()
                c = Card()
                self.assertEqual(type(d.pop_card()),type(c))
```

```
## Test that the return value of the play_war_game function is a tuple
    with three elements, the first of which is a string. (This will
    probably require multiple test methods!)
def test_warInstance(self):
    p = play_war_game()
        self.assertEqual(len(p),3)
        self.assertEqual(type(p[0]), str)

## Write at least 2 additional tests (not repeats of the above
    described tests). Make sure to include a descriptive message in
    these two so we can easily see what you are testing!
def test_myTestInstance1(self):

    self.assertEqual(type(p[0]), str)

def test_myTestInstance2(self):

    self.assertEqual(type(p[0]), str)
```