testA

constanst.py

```
SUITES = ["Hearts","Diamonds","Clubs","Spades"]
VALUES = ["A","2","3","4","5","6","7","8","9","10","J","Q","K"]
OK_MESSAGE = "OK"
```

card.py

```
from constans import VALUES, SUITES, OK_MESSAGE

class Card:
    def __init__(self, suit, value):

        validationResult = self.validateInputValues(suit,value)
        if validationResult != OK_MESSAGE:
            raise Exception(validationResult)

        self.suit = suit
        self.value = value

    def getSuit():
        return self.suit

    def getValue():
        return self.value

    def validateInputValues(self, suit, value):

        if not suit or not isinstance(suit,str) or suit.lower().capitalize() not in SUITES:
            return "Invalid suit"

        if not value or not isinstance(value,str) or value not in VALUES:
            return "Invalid value"

        return OK_MESSAGE
```

deck.py

```
import random
from card import Card

class Deck:
    def __init__(self):
        self.cards = []

    def addCard(self, card):
        self.cards.append(card)

    def shuffleDeck(self):
        random.shuffle(self.cards)

    def displayDeck(self):
        for index, card in enumerate(self.cards):
            print(f"{index+1}° card {card.suit} {card.value}")
```

app.py

```
from deck import Deck
from card import Card
from constans import SUITES, VALUES

def fillPokerDeck():
    pokerDeck = Deck()
    for suite in SUITES:
        for value in VALUES:
            pokerDeck.addCard(Card(suite, value))
    return pokerDeck


def main():
    pokerDeck = fillPokerDeck()
    print("\nstarting Deck:\n")
    pokerDeck.displayDeck()
    pokerDeck.shuffleDeck()
    print("\nAfter Shuffling:\n")
    pokerDeck.displayDeck()

if __name__ == '__main__':
    main()
```

testB

The reported code defines the implementation of some matrix operations (similar to MATLAB).
The function fA, starting from an array, creates a square matrix NxN of zeros.
the function fB creates a lower triangular matrix, setting all the elements below the diagonal to 1.
The function fC calls fA than fB.

Problems:

1.) There is not any validation check on pA value (it must be an integer positive number).
2.) The array A should be passed by reference to fA and fB, or (second option) fA and fB should return the matrix by value.
Modifying the value of a global variable is a bad practice.
3.) "a" is a local variable of fC and it is also a global value, which one is returned by the function?
It depends on the programming language, but generally, reference is always made to the local variable.
It is considered a bad practice to define local variables with the same name as a global variable.