The "Perfect" Shuffle

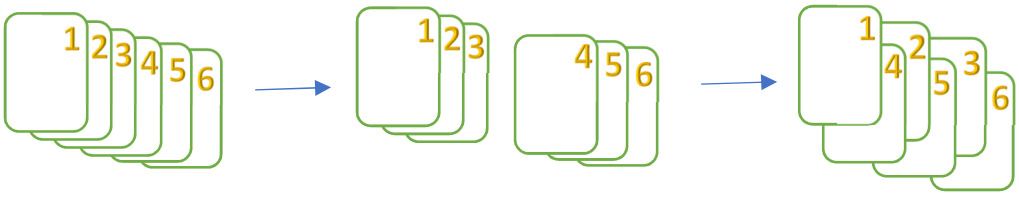
(designed to accompany the AP® Computer Science A Elevens Lab)

- Consider the following questions first:

1. How do you shuffle a deck of cards? What happens to the cards when you shuffle them?
2. If you were to be "perfect" when shuffling a deck of cards, what do you think that means?

- Perfect shuffle:

1. The top half of cards are separated from the bottom half of cards.
2. The original top card stays on top of the pile and all other cards alternate perfectly when shuffled.
3. Below is an example of what happens in a 6-card deck when a perfect shuffle occurs



1st Perfect Shuffle
New: 1, 4, 2, 5, 3, 6

- Expanding the game deck:

1. Now think about a 10-card deck. What would a perfect shuffle look like? What order would the cards finish in?
2. On your own paper, calculate what the deck would look like if you continually performed a perfect shuffle. Record your answers below (you might have to write small, sorry!):

   Order after 1st Shuffle = _____ Order after 5th Shuffle = _____
   Order after 2nd Shuffle = _____ Order after 6th Shuffle = _____
   Order after 3rd Shuffle = _____ Order after 7th Shuffle = _____
   Order after 4th Shuffle = _____ Order after 8th Shuffle = _____

3. How many shuffles did it take to get back to the original deck? Did you notice #1 and #10 never move?!?
4. How many different decks are possible if we only use a perfect shuffle system to shuffle the cards?
5. Mathematics tells us that there should be $10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 3{,}628{,}800$ total decks possible!
6. Open the DeckTester Class from the previous lab. Add the following 6 cards so that the deck now contains a total of 10 cards: E of CAT (10), F of RAT (11), G of BEE (12), H of FOX (13), I of ANT (14), and J of RAM (15).

- Below is the framework for the basic shuffle() method. How would you complete it?

```
public void perfectshuffle(Deck a){                    //Mutator method, Deck parameter
  //code that creates an array called tempCards that will hold 10 card objects
  //code that utilizes a for-loop to copy all of the Deck a cards into the tempCards array
  //code that utilizes the following for-loop (it takes mathematics to discover what is missing
  for(int i=0 ; i<10 ; i++){
    if(i<5)
      this.cards.set(/*MISSING*/,tempCards[i]);
    else
      this.cards.set(/*MISSING*/, tempCards[i]);
  }
```

- Use the DeckTester Class from Lab #4 and add the code below. Verify a perfect sort was done correctly twice!

```
myDeck.perfectshuffle(myDeck);
System.out.println("\n"+myDeck.toString());
myDeck.perfectshuffle(myDeck);
System.out.println("\n"+myDeck.toString());
```