### Lab 4, team 7 (Vincent, Yvette). Go Fish Game

### High-Level Algorithm:

1. **Setup Game:**
   * Prompt the user to input the number of players (including computer).
   * Ask for names of human players.
   * Set up the computer as one of the players.
   * Deal 7 cards to each player if the number of players is 2–4. If there are 5–8 players, deal 5 cards each.
2. **Game Flow:**
   * Shuffle the deck.
   * In each turn:
     + A player (or computer) asks another player for a rank of card.
     + If the other player has cards of that rank, they give them to the asking player.
     + If the other player doesn’t have the requested cards, the asking player "Go Fish" (draws from the deck).
   * Each player checks for sets of 4 cards of the same rank ("books").
   * Player turns continue until the deck is empty or all hands are empty.
   * The game ends, and the player with the most books is the winner.
3. **Classes:**
   * **Card:** Represents a single card with rank and suit.
   * **Deck:** Manages a shuffled deck of cards, including drawing cards.
   * **Player:** Manages a player's hand, including sorting using insertion sort, asking for cards, and checking for books.
   * **GoFishGame:** Manages the flow of the game, including turn-taking, checking game end, and displaying results.

**We will use insertion sort as follows:**

# Sort player's hand using insertion sort

def sort\_hand(self):

for i in range(1, len(self.hand)):

key = self.hand[i]

j = i - 1

while j >= 0 and self.hand[j].rank > key.rank:

self.hand[j + 1] = self.hand[j]

j -= 1

self.hand[j + 1] = key