Submit a file called **Assignment-2.pdf** to the canvas includes **a documentation section** that contains documentation of **smithy, adventurer** cards. The documentation should contain your understanding of the code of smithy and adventurer card. It should also contain documentation of your understanding of the code of **discardCard**() and **updateCoins**() methods. The documentation section should contain statements to
check **preconditions** and **postconditions** of each function. In other words, you need to write what it is required or must be true before the function is called, and what will be true/changed when the function is finished/returned.

Documentation Section:

Smithy - The Smithy costs 4 coins and is an action/kingdom card. The code loops through the drawCard() function three times in order to pull (draw) three new cards for the player from which they can use in their current hand. Then, discardCard() function is then called on the current smithy card which is discarded from the hand.

Adventurer - The Adventurer costs 6 coins and allows the player to pull cards from their deck until they pull the top two treasure cards. The code for this runs a while loop that is based on the condition that less than two treasure cards have been drawn. While there are less than two, the deckCount is checked to see if it needs reshuffling. After that the drawCard() function is called on the top card of the deck. If a treasure card is pulled the drawntreasure count is incremented by 1. If not treasure card is pulled the top card is pulled off and we move to the next card. Once we have incremented the drawnTreasure to the count of two we break the loop and another while loop is entered which will discard each of the non treasure cards that were pulled during the first while loop.

discardCard()- The discardCard() function receives the hand position of the card, the current player, the game state, and a trash indictor. First the trash indicator is checked and if it is not marked then it is placed into the played pile. Next, the played card is set to -1 and checked to see if the player has anymore cards in their hand. The number of cards in hand is reduced by one until the players hand count is 0.

UpdateCoins()- The UpdateCoins() function receives the player name, game state, and bonus amount (if valid). The coin total is set to 0 and then a loop iterates through the player's hand, counters the player's treasure count and each cards value. After the loop, the function adds the bonus coins to the player if applicable.

3- (60 points) Pick five cards implemented in dominion.c. Choose 3 cards of your choice and smithy and adventurer cards are mandatory. Refactor the code so that these cards are implemented in their own functions, rather than as part of the switch statement in cardEffect. You should call the functions for these cards in the appropriate place in cardEffect. Check in your changes, with appropriate git commit messages. Document your changes in the **Assignment-2.pdf** file, under a section called "**Refactor**". Your implementation of at least 4 of these 5 cards should be incorrect in some way, i.e., you should introduce subtle bugs that are hard to catch in your changes. Introducing bugs in smithy and adventurer is mandatory. Write information of your bugs also in a section called **Bugs**. Later in this class, other students will test your code, so try to keep your bugs not superficial. Refactored program should compile without any error.

Refactor/Bugs-

adventurer- I removed the check on if the silver card is drawn during the card drawing process. This will place the silver card in the temp hand, not update the drawn treasure, which makes the silver card not available to play.

Smithy- Instead of a size value of 3. I incremented the value to 4 and placed it in the for loop, so that the card will pull four cards instead of three.

Village- In the village function when the discardCard is called I set the trash flag to 1 instead of 0, so that the card will be trashed (which removes the card permanently), instead of being placed into the discard pile.

Council Room- Instead of the user gaining 4 cards, I depreciated the draw value to 3. I also set the trash flag to 1 instead of 0, similar to the Village, so that the card is trashed (which removes the card permanently) and not placed in the discard pile.

Feast- I set the value of the supply count to be <= 1 instead of 0. This will tell the user that particular supply card is no longer available when there is still one left.