

# Homework 7: Blackjack

## Setup

In this homework, we will be playing the classic casino game of Blackjack. We will accomplish this by creating a single function file with several local functions. The function file should contain the following functions:

1. Blackjack function (main function)
2. Initdeck function
3. Shuffle function
4. Dealcard function
5. Calcscore function
6. Printcard function

The main function should have the header information immediately after the help line.

To play the game, we will make use of a card structure. The card structure is as follows:

- Card Structure
  - Attribute: Suit
    - Type: String value
    - Description: Holds the card's suit
  - Attribute: Value
    - Type: String value
    - Description: Holds the card's value (Ace, 2, 3...)
  - Attribute: Number
    - Type: Numeric value
    - Description: Holds the card's point value (2, 3, 4...)

Blackjack is played like this:

1. The player and dealer each get two cards.
  - a. If the player and dealer's cards add up to 21, then no one wins. This is called a "Push"
  - b. If the player's cards add up to 21, and the dealer's do not, then the player wins. This is "Blackjack"
  - c. If the players' cards add up to less than 21 and the dealer's add up to 21, the player loses. This is also "Blackjack"
  - d. If neither the player's cards, nor the dealer's cards add up to 21, then the game moves on to step 2

2. The player has the option to “Hit” or “Stand”. “Hit” means that they will be dealt another card. “Stand” means that they will keep the cards they have and it will be the dealer’s turn. If they choose to stand, move on to step 4
3. If the user chooses to “Hit” they will be dealt another card. Their score will be calculated.
  - a. If their score is 21 or under, they will have the option to “Hit” or “Stand”
  - b. If their score is above 21, and they have an Ace in their hand, their score is decreased by 10. This is the duality of the Ace card. They will again have the option to “Hit” or “Stand”
  - c. If their score is above 21, and they don’t have any Aces in their hand, or their score was already decreased and they are still above 21, then they “Bust” and lose the game.
  - d. If their score is under 21 and they have 5 cards in their hand, they win. This is called a “Charlie”
4. If the user chooses to “Stand”, then the dealer begins their turn.
5. The dealer’s score is calculated
  - a. If the dealer’s score is less than 17, the dealer MUST “Hit”
  - b. If the dealer’s score is 17 or above, the dealer will “Stand”
6. The dealer’s score is recalculated
  - a. If the dealer’s score is 21 or under, move on to step 7
  - b. If the dealer’s score is over 21, the dealer “Busts” and the player wins
  - c. If the dealer’s score is equal to the player’s score, then no one wins. This is also a “Push”
7. If the dealer’s score and the player’s score are 21 or under, but are not the same, the person with the higher score wins the game.

## Part 1: initdeck() Function

The initdeck() function has only one input and only one output. The input of this function is a single positive integer value. This value represents the number of decks that the function will be creating. The output of this function is a structure array of cards. The number of cards in the structure array is dependent upon the number of decks to create. The function should operate like this:

1. Check the input:
  - a. It should be a **single, positive, integer value**. If it’s not, return an error to the user.
2. Create a single card structure with the correct fields as described in the setup
3. Replicate the card structure 52 times to form a single deck
4. Fill each card with the correct values for suit
5. Fill each card with the correct values for value
6. Fill each card with the correct values for number:
  - a. Cards 2 – 10 have values 2 – 10 accordingly
  - b. Cards Jack – King have a values of 10
  - c. Aces have a value of 11
7. Replicate the card structure array (the deck) the correct number of times to create multiple decks
8. Return the card structure array as output

## Part 2: shuffle() Function

The shuffle() function has only one input and only one output. The input of this function is a **card structure array**. The output of this function is a card structure array. The input is the deck(s) of cards you want to shuffle. The output is the shuffled deck(s). Your code should perform the following:

1. Check the input:
  - a. It should be a **structure array** with at least 52 elements (cards). If it's not, return an error to the user.
2. Create two random integers with a maximum value equal to the number of elements of the card structure array
3. Swap the cards of the structure array at these two random integer indexes (swap #45 and #27 for example)
4. Repeat steps 2 and 3 at least 3000 times
5. Return the shuffled structure array as output

## Part 3: dealcard() Function

The dealcard() function has only one input and two outputs. The input of this function is the **card structure array** (deck) from which you are dealing the card. The outputs from this function are:

1. The card that was dealt
2. The card structure array with the card that was dealt removed

The function should operate like this:

1. Check the input:
  - a. It should be a **structure array**
2. Output the first card as the first output
3. **Delete** the first card from the deck
4. Output the new deck as the second output

## Part 4: calcscore() Function

The calcscore() function has only one input and only one output. The input of this function is a card structure array (the cards that either the dealer or the player has). The output of this function is a single integer value that represents the total value of the structure array (either the player's score, or the dealer's score). The function should operate like this:

1. Check the input:
  - a. It should be a **structure array**
2. Calculate the raw score (this is the sum of the number field of each card)
3. Calculate the number of Aces in the hand

4. If the score is above 21 and the number of Aces is at least 1, decrease the raw score by 10 and the number of Aces by 1 until either the raw score is 21 or below or the number of Aces is less than 1.
5. Output the score as the output

## Part 5: printcard() Function

The printcard() function has only one input and no outputs. The input of this function is a **single card structure**. The function should operate like this:

1. Check the input:
  - a. It should be a **single card structure**
2. Using an fprintf() statement, print the card's value and suit in the following way:

10 of Diamonds  
Jack of Clubs

## Part 6: blackjack() Function

The blackjack() function is the main function. It has only one input and no output. The input of the function is a single integer value which represents the number of decks that you will be playing with. The function should operate like this:

1. Check the input;
  - a. It should be a single, positive, integer value. If it is not, produce an error
2. Create the deck(s)
3. Shuffle the deck(s)
4. Deal two cards to the player's hand
  - a. Keep track of the number of cards for the player
5. Deal two cards to the dealer's hand
  - a. Keep track of the number of cards for the dealer
6. Print the cards for the player
7. Print the first card for the dealer (the dealer never reveals his second card until it's his turn)
8. Print the score for the player
9. Determine if anyone has reached "Blackjack"
  - a. If so, announce the winner and end the function
10. Determine if there was a "Push"
  - a. If so, announce the push and end the function
11. Ask the player to hit or stand
12. If the player hits, deal the player a card
  - a. Print the card that was dealt
  - b. Update the number of cards the player has
  - c. Calculate the player's new score
  - d. If the player's score is above 21, they lose. Announce the loser and end the function

- e. If the player's score is 21 or below, and they have 5 cards, they win. Announce the winner and end the function
  - f. If the player's score is 21 or below and they have less than 5 cards, go back to Step 11
13. If the player stands, move on to the dealer
14. Print both of the dealer's cards
15. Print the dealer's score
16. If the dealer's score is below 17, hit
- a. Print the card that was dealt
  - b. Update the number of cards the dealer has
  - c. Calculate the dealer's new score
  - d. If the dealer's score is above 21, they lose. Announce the loser and end the function
  - e. If the dealer's score is 21 or below, and they have 5 cards, they win. Announce the winner and end the function
  - f. If the dealer's score is 21 or below and they have less than 5 cards, go back to Step 16.
17. Check to see who won
18. Announce the winner, then end the function.

## Sample Output

Below is the sample output. User input is highlighted in **red**:

```
>> blackjack(2)
Player has:
5 of Hearts
3 of Hearts
Player score: 8
Dealer showing:
4 of Hearts
Choose from the following:
    1) Hit
    2) Stand
Choose: 1
Player was dealt: Queen of Clubs
Player score: 18
Player has:
5 of Hearts
3 of Hearts
Queen of Clubs
Choose from the following:
    1) Hit
    2) Stand
Choose: 2
Dealer has:
4 of Hearts
```

7 of Hearts

Dealer score: 11

Dealer was dealt: 4 of Clubs

Dealer score: 15

Dealer has:

4 of Hearts

7 of Hearts

4 of Clubs

Dealer was dealt: 8 of Hearts

Dealer score: 23

Bust! Player wins!

>> blackjack(1)

Player has:

7 of Diamonds

Queen of Spades

Player score: 17

Dealer showing:

2 of Diamonds

Choose from the following:

1) Hit

2) Stand

Choose: 3

Invalid choice!

Choose from the following:

1) Hit

2) Stand

Choose: 2

Dealer has:

2 of Diamonds

3 of Clubs

Dealer score: 5

Dealer was dealt: 3 of Spades

Dealer score: 8

Dealer has:

2 of Diamonds

3 of Clubs

3 of Spades

Dealer was dealt: Queen of Hearts

Dealer score: 18

Dealer has:

2 of Diamonds

3 of Clubs

3 of Spades  
Queen of Hearts  
Dealer wins!

```
>> blackjack(2.1)
Error using blackjack (line 7)
Input must be integer value
```

```
>> blackjack(1)
Player has:
10 of Diamonds
Ace of Hearts
Player score: 21
Dealer showing:
8 of Clubs
Blackjack! Player wins!
```

```
>> blackjack(1)
Player has:
9 of Hearts
8 of Hearts
Player score: 17
Dealer showing:
5 of Diamonds
Choose from the following:
    1) Hit
    2) Stand
Choose: 1
Player was dealt: Ace of Hearts
Player score: 18
Player has:
9 of Hearts
8 of Hearts
Ace of Hearts
Choose from the following:
    1) Hit
    2) Stand
Choose: 1
Player was dealt: 10 of Hearts
Player score: 28
Bust! Dealer wins!
```

## Grading

Item	Points
Part 1: Initdeck()	15
Part 2: Shuffle()	10
Part 3: Dealcard()	10
Part 4: Calcscore()	15
Part 5: Printcard()	10
Part 6: Blackjack()	30
<b>Total*</b>	<b>90</b>

\*Grading note: Points will be taken off for poor code style including: not using camel case, not using descriptive variable names, not formatting output correctly, not neatly displaying output, no comments, no header. Points will also be taken off for incorrect submission of files, including incorrect names of files and zip files.

## Deliverables

This assignment will be due:

Friday, March 30, 2018 at 11:59:59pm

Submitted through Blackboard

Compressed in a zipped folder HW7.zip