In this assignment you will write a Matlab script file that simulates a simple version of the card game of Blackjack, or 21. The computer will play the role of the dealer, and you will be the only player. The objective of the game is to accumulate a higher score than the dealer, but without exceeding 21. Your score is computed by adding the values of your individual cards. For this assignment the *card value* is determined by either:

- The number on the card (2 through 10);
- Ten if it is a "face" card (Jack, Queen, or King); or
- One if it is an Ace. (Here we simplify the regular rule in which an Ace can count for either one or 11.)

Note that the suit is irrelevant.

The game consists of the following steps:

1. The dealer draws two cards, but reveals only one of them to you.
2. You draw two cards. (In the actual game you hide one from the other players, but that is unnecessary here since you are the only player.)
3. You choose whether or not to take another card: *hit* means yes and *stand* means no. This decision should be based on the value of your cards and what the dealer shows. If you hit and your total (sum of the card values) is more than 21, then you *bust*, and the dealer wins. This step then repeats until you either bust or decide to stand.
4. It is now the dealer's turn provided that you did not bust in the preceding step. The dealer reveals its hidden card, and then chooses to hit or stand based on a fixed rule: If the total card value is less than 17, then the dealer hits. Otherwise, the dealer stands. If the dealer hits and the total is more than 21, then you win! This step repeats until the dealer busts or stands.
5. If both you and the dealer stand (that is, neither you nor the dealer has busted in the preceding two steps), then there is a "showdown" in which you and the dealer compare your scores to determine the winner.

Write Matlab statements corresponding to each of the preceding steps with the following structure:

1. Using the Matlab function `randi`, assign the dealer two random card values, compute the total value, and print out one of the values. Assume that each card is equally likely to be chosen from the set: { `Ace, 2, ⋯, 10, Jack, Queen, King` }, so that the value is between 1 and 10. (This assumes that the dealer draws the cards from many – actually an infinite number – of shuffled decks. That is, drawing any particular card does not reduce the chances of drawing that card again.) One way to implement this choice is to number the cards in this set from 1 to 13, and set the card values for cards 11 through 13 (Jack, Queen, King) to 10. An easy way to do this is with the `min` command.

2. Similarly, assign the player two card values, print those values, and compute the total.
3. Write a loop for the player, which contains the following statements:
   (a) Prompt the player to enter hit (1) or stand (0).
   (b) If the player hits then assign another card value, and compute and print the total.
   (c) If the total exceeds 21, then print that the player busts (loses).

   Note that the program exits the loop if either the player busts or stands.
4. If the player has not busted, then it is the dealer's turn. In that case, print the card values for the dealer.
5. Write a loop for the dealer, which contains the following statements:
   (a) If the dealer's total is less than 17, then assign another card to the dealer, print the new card value and the total.
   (b) If the dealer's total is greater than 21, indicate that the dealer busts and the player wins.

   As for the player, the program will exit this loop if the dealer either busts or stands.
6. If neither the player nor the dealer has busted, then compare and print their totals along with a message stating who wins. Include the possibility of a tie.

Run your program as the player and copy and paste outputs corresponding to four outcomes:

1. The dealer busts and you win;
2. You bust and lose;
3. You and the dealer stand, and you win;
4. You and the dealer tie. This is likely to take several trials.

An example program run corresponding to each of these outcomes is shown below.

Example Program Outputs:

```
>> hw3_v2
The dealer shows 10 and hides the other card.
You show 8 and hide 1, totaling 9
Hit (1) or stand (0)? 1
You drew a 1, totaling 10
Hit (1) or stand (0)? 1
You drew a 7, totaling 17
Hit (1) or stand (0)? 0
The dealer has 10 and 5, totaling 15.
The dealer draws a 10, totaling 25.
The dealer busts!
You win! Hooray!

>> hw3_v2
The dealer shows 5 and hides the other card.
You show 3 and hide 8, totaling 11
Hit (1) or stand (0)? 1
You drew a 4, totaling 15
Hit (1) or stand (0)? 1
You drew a 9, totaling 24
You bust!
Sorry, you lose.

>> hw3_v2
The dealer shows 3 and hides the other card.
You show 9 and hide 7, totaling 16
Hit (1) or stand (0)? 1
You drew a 5, totaling 21
Hit (1) or stand (0)? 0
The dealer has 3 and 4, totaling 7.
The dealer draws a 10, totaling 17.
The dealer stands:
The dealer has 17, and you have 21.
You win! Hooray!

>> hw3_v2
The dealer shows 10 and hides the other card.
You show 2 and hide 2, totaling 4
Hit (1) or stand (0)? 1
You drew a 4, totaling 8
Hit (1) or stand (0)? 1
You drew a 10, totaling 18
Hit (1) or stand (0)? 0
The dealer has 10 and 8, totaling 18.
The dealer stands:
The dealer has 18, and you have 18.
You tied!
```