MCA 103 - Assignment 2 **MCA First Year 2017**

Date of Issue 23rd September 2017 **Due Date 11th October 2017**

General Instructions

- Programming language to be used is R for every aspect of calculation. No manual calculation should be done
- Do it in groups of one/two/three but not in more than three
- Please give all numerical answers to 6 digits of precision.
- CR will collect all the folders of all the groups in a pen-drive and submit it to me for evaluation maximum by 4pm.
- You may also refer to e-books for syntax of specific commands.
- **This assignment is optional for students participating in Open Data Hackathon, however they can help the other members of the class or can also submit their own solutions which will be a part of the final evaluation of the Internal Assessment.**

Programming Instructions

- Note that all your programs should have proper alignment, indentation and proper comments.
- All constants / variables / functions etc. should have meaningful names.
- Overall, programs should be readable. If programs fails to execute in R, you will get zero for everything.
- Submission files – TwentyOne.R, 21readme.txt
- Read me files should give information about code, functions and data structures used, diagrammatic representation of the concepts, etc. You may refer to preparation of readme file from here.

Description

You are going to create a simple card game of 21. In this game, the single player has to try to come as close to a score of 21 without going over and beat the computer's score.

1. Set up an array called “PlayingCard” that is having a string representing card’s rank and suit. For example, the card with rank 1 and suit 'S' is "Ace of Spades".

2. Set up a routine called “DeckOfPlayingCards” which contains the following information:

INSTANCE VARIABLES

- A list of 52 PlayingCard having their rank and suit specified
- Index of the "top" card in the deck

FUNCTIONS

INIT - initializes each PlayingCard tuple to a unique card in a standard deck.

SHUFFLE - randomly permute the order of the cards in the array and sets the top card to the first card in the array. In order to shuffle the cards, you may use the following algorithm:

for i <-- 0 to 50
pick a random integer j between i and 51
swap card i with card j
end for

DEAL - returns the top card in the deck and sets the top card index to the next card

3. Set up a routine “TwentyOne” which acts a main method that creates the deck of cards and then allows the user to play the Game of 21.

Rules of the Game

Each card has a point value based on its rank (the suit is ignored in this game). The cards with ranks 2 through 10 have point values of 2 through 10 respectively. The "face" cards (Jack, Queen, King) have a point value of 10 each. The Ace is considered as 11 points, unless that puts the player over a total of 21 points, in which case it reverts to 1 point instead. For example, the following cards are dealt to the player and the total scores are shown to their right:

CARD	CARD SCORE	TOTAL SCORE
5 of Diamonds	5	5
Ace of Hearts	11	16
2 of Clubs	2	18
Ace of Spades	1	19
2 of Hearts	4	21

In each game, the deck of cards is shuffled, and the user starts with the ***first two cards*** of the deck. The user may pick the next card of the deck by inputting "HIT" or the user may stop at this point by inputting "STAY". The user can pick as many cards as he or she wants in order to try to come up with a score as close to 21 without going over. If the user goes over 21 points, the user automatically loses and the computer wins. Otherwise, if the user stops with a total score less than or equal to 21, then the computer plays. The computer starts with the next two cards of the deck. The computer automatically "HITS" until its score is at least 17. If the computer goes over 21 (but the user did not), then the user wins automatically. Otherwise, the winner is the player with the higher score. A tie (same total score) is won by the computer.

Input Processing

Input will come from the keyboard in this game. The user should input "HIT" or "STAY" as the game proceeds. Any other input should flag an error "Unrecognized input", etc. and you should ask for the input again. At the end of the game, you should ask the user if he/she wants to play again. The input here will be "Y" or "N" (lowercase ok). All other input will lead to an error and you should ask the user to input again. See OUTPUT PROCESSING for an example of correct input.

Output Processing

LET'S PLAY 21!
SHUFFLING CARDS...

YOUR TURN
5 of Diamonds 5
Ace of Hearts 16
HIT or STAY? *HIT*
7 of Clubs 13
HIT or STAY? *HIT*
3 of Clubs 16
HIT or STAY? *HIT*
Ace of Spades 17

HIT or STAY? *HIT*

4 of Hearts 21

HIT or STAY? *STAY*

COMPUTER'S TURN

King of Clubs 10

9 of Diamonds 19

YOUR SCORE: 21

COMPUTER'S SCORE: 19

YOU WIN!

PLAY AGAIN? (Y/N) *Y*

LET'S PLAY 21!

SHUFFLING CARDS...

YOUR TURN

8 of Hearts 8

Queen of Spades 18

HIT or STAY? *STAY*

COMPUTER'S TURN

Jack of Hearts 10

3 of Clubs 13

7 of Spades 20

YOUR SCORE: 18

COMPUTER'S SCORE: 20

YOU LOSE!

PLAY AGAIN? (Y/N) *Y*

LET'S PLAY 21!

SHUFFLING CARDS...

YOUR TURN

Queen of Hearts 10

5 of Diamonds 15

HIT or STAY? *HIT*

7 of Hearts 22

YOU LOSE!

PLAY AGAIN? (Y/N) *Y*

LET'S PLAY 21!

SHUFFLING CARDS...

YOUR TURN

4 of Spades 4

10 of Clubs 14

HIT or STAY? *HIT*

6 of Hearts 20

HIT or STAY? *STAY*

COMPUTER'S TURN

5 of Hearts 5

Jack of Spades 15

Ace of Diamonds 16

4 of Clubs 20

YOUR SCORE: 20

COMPUTER'S SCORE: 20

YOU LOSE!

PLAY AGAIN? (Y/N) *Y*

LET'S PLAY 21!

SHUFFLING CARDS...

YOUR TURN

8 of Hearts 8

9 of Diamonds 17

HIT or STAY? *STAY*

COMPUTER'S TURN

King of Clubs 10

6 of Spades 16

Jack of Diamonds 26

YOU WIN!

PLAY AGAIN? (Y/N) *N*

Extra Credits (Optional)

Guide the user for Hit and Stay by telling the probability of getting 21 if they HIT the deck again based on their current cards.

All the Best