**CSE 231** **Fall 2020**

# Computer Project #10

## Assignment Overview

This assignment focuses on the design, implementation and testing of a Python program which uses an instructor-supplied module to play a card game, as described below.

It is worth 55 points (5.5% of course grade) and must be completed no later than 11:59 PM on Wednesday, December 2, 2020 .

## Assignment Deliverables

The deliverable for this assignment is the following file:

**`proj10.py`** – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir system** before the project deadline.

## Assignment Background

Scorpion is a solitaire card game which is played by one person with a standard 52-card deck of cards. The rules and a tutorial video are available at:

http://worldofsolitaire.com/

Click on "Choose Game" at the top of the screen and click on "Scorpion".

Your program will allow the user to play a simplified version of Scorpion, with the program managing the game. The game rules are given below, but you should play the game online to understand it before you try to code it.

## Game Rules

1. Start: The game is played with one standard deck of 52 cards. The deck is shuffled and becomes the initial *stock* (pile of cards). The cards in the stock are placed face down.

Forty-nine cards (all but three) are then dealt from the stock, left to right, one to each column of a seven-column *tableau*, and then to successive rows in the columns. You need to continue dealing an additional card to each column, until each column has seven cards. The order of the initial deal is important to match tests. All tableau cards are visible except the first three rows of cards in the first three columns, i.e. the upper-left nine cards are face down.

The game also has a *foundation*, which has four piles of cards. The foundation starts out empty and is filled during play.

2. Goal: The game is won when the foundation is full.

3. Moves: A player can move one card or a sequence of cards at a time, and the moved card or sequence must be moved to the bottom card of a tableau column. The card or the top card of a sequence can be moved to a card at the bottom of a tableau column, if they have the same suit and the destination card has a rank one higher than the source card. If a column is empty, only a king (and the sequence attached to it) can be placed there. If a card is face down, it cannot be moved.

Once a column has the entire sequence of a suit (king down to ace), that sequence is automatically moved to an empty foundation slot. (For testing consistency in our program the foundation will be filled left to right.)

No other moves are permitted.

4. Deal: A player can choose to deal instead of moving a card. The stock has only three cards and they are dealt left to right on the leftmost three columns.

**Assignment Specifications**

You will develop a program that allows the user to play Scorpion according to the rules given above. The program will use the instructor-supplied **`cards.py`** module to model the cards and deck of cards. To help clarify the specifications, we provide a sample interaction with a program satisfying the specifications at the end of this document.

1. The program will recognize the following commands (upper or lower case):

```
D: Deal to the Tableau (one card to first three columns).
M c r d: Move card from Tableau column, row to end of column d.
R: Restart the game (after shuffling)
H: Display the menu of choices
Q: Quit the game
```

where **`c,r`** and **`d`** denote column numbers, and the columns are numbered from 1 to 7.

The program will repeatedly display the current state of the game and prompt the user to enter a command until the user wins the game or enters "**`Q`**", whichever comes first.

The program will detect, report and recover from invalid commands. None of the data structures representing the stock, tableau, or foundation will be altered by an invalid command.

2. The program will use the following function to initialize a game:

**`initialize() → (stock, tableau, foundation)`**

That function has no parameters. It creates and initializes the stock, tableau, and foundation, and then returns them as a tuple, in that order. This corresponds to rule 1 in the game rules:

- `stock` is a deck class object,
- `foundation` is an empty list of four lists, i.e. `[[],[],[],[]]`
- `tableau` is a list of seven lists, each containing seven of the dealt cards. The first element of `tableau` is the leftmost column when displayed.

The deck is shuffled and becomes the initial `stock` (pile of cards). Forty-nine cards are then dealt from the stock, left to right, one to each column of a seven-column `tableau`, and then to successive rows in the columns. The order of dealing is important to match tests.

3. The program will use the following function to deal cards to the tableau:

**`deal_from_stock(stock,tableau): → None`**

That function has two parameters: the data structure representing the stock and the data structure representing the tableau. It will deal a card from the stock to the leftmost three columns of the tableau. It will always deal three cards. If the stock is empty, do not deal any cards to the tableau.

4. The program will use the following function to display the current state of the game:

**`display( stock, tableau, foundation ) → None`**

Provided.

5. The program will use the following function to prompt the user to enter an option and return a representation of the option designed to facilitate subsequent processing.

**`get_option()→ list`**

Provided

That function takes no parameters. It prompts the user for an option and checks that the input supplied by the user is of the form requested in the menu. Valid inputs for options are described in item (bullet) 1 of the specifications. If the input is not of the required form, the function prints an error message and returns `None`.

Note that input with the incorrect number of arguments, e.g. M 2, or incorrect types, e.g. F D, will return `None.`

The function returns a list as follows:

- **`None`**, if the input is not of the required form
- **`['D']`**, for deal
- **`['M', c, r, d]`**, where **`c, r`** and **`d`** are **`int`**'s, for moving a card (or sequence) within the tableau from **`(c,r)`** to column **`d`**
- **`['R']`**, for restart
- **`['H']`**, for displaying the menu
- **`['Q']`**, for quit

6. The program will use the following function to determine if a requested move is valid:

**`validate_move(tableau,src_col,src_row,dst_col) → bool`**

That function has four parameters: the data structure representing the tableau, two **`ints`** indicating the column and row for the source card (and sequence), and the **`int`** for the destination column where the source card (and sequence) should be moved. The function will return **`True,`** if the move is valid; and **`False`**, otherwise. This function does not display anything.

There are two main rules to consider:

- An empty column can only have a king (and its sequence) moved to it.
- A card (and its sequence) can only be moved to a card of the same suit and whose rank is one greater

Also, remember to check that the source card exists at the specified column and row, e.g. the index may be out of range; `try-except` works well for that.

7. The program will use the following function to move a card within the tableau:

**`move(tableau,src_col,src_row,dst_col) → Boolean`**

That function has four parameters: the data structure representing the tableau, the column and row of the source card, and the destination column. If the move is valid (determined by calling `validate_move`), the function will update the tableau; otherwise, it will do nothing to it. Note that slicing is useful to create the sequence to be moved (and what is left in the source list) and that the list `extend` method will likely be useful. Remember that if a face-down card is now at the bottom of a column, it must be flipped to be face up.

8. The program will use the following function to check if the game has been won:

**`check_for_win( foundation ) → None`**

That function checks to see if the foundation is full. It returns **`True`**, if the foundation is full and **`False`**, otherwise.

9. The program will use the following function to check if a column in the tableau is a complete sequence from king down to ace of the same suit:

**`check_sequence( column_lst ) → Boolean`**

It returns **`True`**, if the sequence is complete and **`False`**, otherwise.
**Hint:** think of this function as finding out if the sequence is not complete. That is, if cards are not in the same suit return **`False`**; if adjacent cards' ranks do not differ by one, return **`False`**. If at the end you haven't returned **`False`**, then simply return **`True`**.

10. The program will use the following function to check if any column sequences are complete (call **`check_sequence`**) and if so, move it to the foundation:

**`move_to_foundation( tableau,foundation )→ None`**

Fill the foundation from left to right; one complete sequence to each foundation slot.

11. Once you write all your function, it is time to write your **`main`** function:
    a) Your program should start by initializing the board (the stock, the tableau and the foundation).
    b) Display the menu and the board.
    c) Ask to input an option and check the validity of the input.
    d) If `'D'`, deal three cards from the stock to the tableau. Display the board.
    e) If `'M c r d'`, move card (sequence) within Tableau `(c,r)` to column `d`. The integers c, r, and d are values between 1 and 7, but remember that lists start from index 0.
    If a move is successful, check to see if any sequences are complete to be automatically moved to the foundation. Then check to see if the user won; if so, restart. Display the board.
    f) If `'R'`, restart the game by initializing the board (after shuffling). Display the board.
    g) If `'H'`, display the menu of choices
    h) If `'Q'`, quit the game
    i) If none of these options, the program should display an error message.
    j) The program should repeat until the user quit the game. Display a goodbye message. If the player wins the game, the program automatically restarts another game.

**Assignment Notes**

1. Before you begin to write any code, play with the provided demo program and look over the sample interaction supplied on the project website to be sure you understand the rules of the game and how you will simulate the demo program. The demo program is at http://worldofsolitaire.com/: Click on "Choose Game" at the top of the screen and click on "Scorpion".

2. We provide a module called **`cards.py`** that contains a Card class and a Deck class. Your program must use this module (**`import cards`**). *Do not modify this file!* This is a generic module for any card game. Your program must implement "Scorpion" *without modifying the* **`cards`** *module***.**

3. Laboratory Exercise #10 demonstrates how to use the **`cards`** module. Understanding those programs should give you a good idea how you can use the module in your game.

4. **We have provided a framework named `proj10.py` to get you started.**
***Using this framework is mandatory*. Begin by downloading `proj10.py`. Check that it compiles. Gradually replace the "stub" code (marked with comments) with your own code. (Delete the stub code.)**

5. The coding standard for CSE 231 is posted on the course website:

http://www.cse.msu.edu/~cse231/General/coding.standard.html

Items 1-9 of the Coding Standard will be enforced for this project.

6. Your program may not use any global variables inside of functions. That is, all variables used in a function body must belong to the function's local name space. The only global references will be to functions and constants.

7. Your program may not use any nested functions. That is, you may not nest the definition of a function inside another function definition.

8. Your program must contain the functions listed above; you may develop additional functions, as appropriate.

**TEST 1 (incomplete game)**

```
Welcome to Scorpion Solitaire

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦ 10♦
 2  XX  XX  XX  9♥  4♠  6♣  J♠
 3  XX  XX  XX  9♦  2♦  7♦ 10♥
 4  5♥  A♠  5♣  7♣ 10♠  A♥  8♦
 5  6♥  6♠  K♣  8♥  5♦  9♣  3♠
 6  K♥  J♥  3♣  A♣  K♠  9♠  Q♦
 7  5♠  J♣  3♦  2♣  J♦  8♠  2♠


Input options:
    D: Deal to the Tableau (one card to first three columns).
    M c r d: Move card from Tableau (column,row) to end of column d.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
    Q: Quit the game


Input an option (DMRHQ): M 7 1 5

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦
 2  XX  XX  XX  9♥  4♠  6♣
 3  XX  XX  XX  9♦  2♦  7♦
 4  5♥  A♠  5♣  7♣ 10♠  A♥
 5  6♥  6♠  K♣  8♥  5♦  9♣
 6  K♥  J♥  3♣  A♣  K♠  9♠
 7  5♠  J♣  3♦  2♣  J♦  8♠
 8                  10♦
 9                   J♠
10                  10♥
11                   8♦
12                   3♠
13                   Q♦
14                   2♠


Input an option (DMRHQ): M 2 5 7
Error in move: M , 2 , 5 , 7

Input an option (DMRHQ): M 3 5 7

stock   foundation
 XX
```

```
tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦  K♣
 2  XX  XX  XX  9♥  4♠  6♣  3♣
 3  XX  XX  XX  9♦  2♦  7♦  3♦
 4  5♥  A♠  5♣  7♣ 10♠  A♥
 5  6♥  6♠      8♥  5♦  9♣
 6  K♥  J♥      A♣  K♠  9♠
 7  5♠  J♣      2♣  J♦  8♠
 8                  10♦
 9                   J♠
10                  10♥
11                   8♦
12                   3♠
13                   Q♦
14                   2♠


Input an option (DMRHQ): M 5 10 2
Error in move: M , 5 , 10 , 2

Input an option (DMRHQ): M 5 2 1

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦  K♣
 2  XX  XX  XX  9♥      6♣  3♣
 3  XX  XX  XX  9♦      7♦  3♦
 4  5♥  A♠  5♣  7♣      A♥
 5  6♥  6♠      8♥      9♣
 6  K♥  J♥      A♣      9♠
 7  5♠  J♣      2♣      8♠
 8  4♠
 9  2♦
10 10♠
11  5♦
12  K♠
13  J♦
14 10♦
15  J♠
16 10♥
17  8♦
18  3♠
19  Q♦
20  2♠


Input an option (DMRHQ): M 1 9 7

stock   foundation
 XX

tableau
```

```
    1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦  K♣
 2  XX  XX  XX  9♥      6♣  3♣
 3  XX  XX  XX  9♦      7♦  3♦
 4  5♥  A♠  5♣  7♣      A♥  2♦
 5  6♥  6♠      8♥      9♣ 10♠
 6  K♥  J♥      A♣      9♠  5♦
 7  5♠  J♣      2♣      8♠  K♠
 8  4♠                      J♦
 9                         10♦
10                          J♠
11                         10♥
12                          8♦
13                          3♠
14                          Q♦
15                          2♠


Input an option (DMRHQ): D

stock   foundation
 XX

tableau
    1   2   3   4   5   6   7
 1  XX  XX  XX  Q♣  7♠  K♦  K♣
 2  XX  XX  XX  9♥      6♣  3♣
 3  XX  XX  XX  9♦      7♦  3♦
 4  5♥  A♠  5♣  7♣      A♥  2♦
 5  6♥  6♠  Q♥  8♥      9♣ 10♠
 6  K♥  J♥      A♣      9♠  5♦
 7  5♠  J♣      2♣      8♠  K♠
 8  4♠  3♥                  J♦
 9  6♦                     10♦
10                          J♠
11                         10♥
12                          8♦
13                          3♠
14                          Q♦
15                          2♠

Input an option (DMRHQ): H

Input options:
    D: Deal to the Tableau (one card to first three columns).
    M c r d: Move card from Tableau (column,row) to end of column d.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
    Q: Quit the game


Input an option (DMRHQ): R

stock   foundation
 XX

tableau
```

```
    1   2   3   4   5   6   7
 1  XX  XX  XX 10♠  4♣  K♥  3♦
 2  XX  XX  XX  Q♥  9♦  8♥  2♠
 3  XX  XX  XX  K♦  9♠  7♣  7♦
 4  8♦  A♦  K♠  8♣  A♥ 10♦  3♠
 5  4♥  4♦  7♠  4♠  3♥  5♣  2♣
 6  J♣  6♣  A♣  K♣  J♠  Q♣  5♦
 7  2♦  6♠  J♥  2♥  5♠  Q♦  3♣
```

Input options:
    D: Deal to the Tableau (one card to first three columns).
    M c r d: Move card from Tableau (column,row) to end of column d.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
    Q: Quit the game

Input an option (DMRHQ): q
Thank you for playing.

**TEST 2:**

```
Welcome to Scorpion Solitaire.


stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  7♠  7♦
 2  XX  XX  XX  6♥  6♣  6♠  6♦
 3  XX  XX  XX  5♥  5♣  5♠  5♦
 4  Q♠  J♦  J♠  4♥  4♣  4♠  4♦
 5  Q♦ 10♦ 10♠  3♥  3♣  3♠  3♦
 6  K♦  9♦  9♠  2♥  2♣  2♠  2♦
 7  K♠  8♦  8♠  A♥  A♣  A♠  A♦


Input options:
    D: Deal to the Tableau (one card to first three columns).
    M c r d: Move card from Tableau (column,row) to end of column d.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
    Q: Quit the game


Input an option (DMRHQ): M 7 1 2

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  7♠
 2  XX  XX  XX  6♥  6♣  6♠
 3  XX  XX  XX  5♥  5♣  5♠
 4  Q♠  J♦  J♠  4♥  4♣  4♠
 5  Q♦ 10♦ 10♠  3♥  3♣  3♠
 6  K♦  9♦  9♠  2♥  2♣  2♠
 7  K♠  8♦  8♠  A♥  A♣  A♠
 8      7♦
 9      6♦
10      5♦
11      4♦
12      3♦
13      2♦
14      A♦


Input an option (DMRHQ): M 6 1 3
```

```
stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣
 2  XX  XX  XX  6♥  6♣
 3  XX  XX  XX  5♥  5♣
 4  Q♠  J♦  J♠  4♥  4♣
 5  Q♦ 10♦ 10♠  3♥  3♣
 6  K♦  9♦  9♠  2♥  2♣
 7  K♠  8♦  8♠  A♥  A♣
 8      7♦  7♠
 9      6♦  6♠
10      5♦  5♠
11      4♦  4♠
12      3♦  3♠
13      2♦  2♠
14      A♦  A♠


Input an option (DMRHQ): M 1 7 6

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♠
 2  XX  XX  XX  6♥  6♣
 3  XX  XX  XX  5♥  5♣
 4  Q♠  J♦  J♠  4♥  4♣
 5  Q♦ 10♦ 10♠  3♥  3♣
 6  K♦  9♦  9♠  2♥  2♣
 7      8♦  8♠  A♥  A♣
 8      7♦  7♠
 9      6♦  6♠
10      5♦  5♠
11      4♦  4♠
12      3♦  3♠
13      2♦  2♠
14      A♦  A♠


Input an option (DMRHQ): M 1 7 7
Error in move: M , 1 , 7 , 7

Input an option (DMRHQ): M 1 6 7

stock   foundation
 XX
```

```
tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♠  K♦
 2  XX  XX  XX  6♥  6♣
 3  XX  XX  XX  5♥  5♣
 4  Q♠  J♦  J♠  4♥  4♣
 5  Q♦ 10♦ 10♠  3♥  3♣
 6      9♦  9♠  2♥  2♣
 7      8♦  8♠  A♥  A♣
 8      7♦  7♠
 9      6♦  6♠
10      5♦  5♠
11      4♦  4♠
12      3♦  3♠
13      2♦  2♠
14      A♦  A♠


Input an option (DMRHQ): M 1 5 7

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♠  K♦
 2  XX  XX  XX  6♥  6♣      Q♦
 3  XX  XX  XX  5♥  5♣
 4  Q♠  J♦  J♠  4♥  4♣
 5     10♦ 10♠  3♥  3♣
 6      9♦  9♠  2♥  2♣
 7      8♦  8♠  A♥  A♣
 8      7♦  7♠
 9      6♦  6♠
10      5♦  5♠
11      4♦  4♠
12      3♦  3♠
13      2♦  2♠
14      A♦  A♠


Input an option (DMRHQ): M 1 4 6

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♠  K♦
 2  XX  XX  XX  6♥  6♣  Q♠  Q♦
 3  K♥  XX  XX  5♥  5♣
 4      J♦  J♠  4♥  4♣
```

```
 5     10♦ 10♠  3♥  3♣
 6      9♦  9♠  2♥  2♣
 7      8♦  8♠  A♥  A♣
 8      7♦  7♠
 9      6♦  6♠
10      5♦  5♠
11      4♦  4♠
12      3♦  3♠
13      2♦  2♠
14      A♦  A♠
```

Input an option (DMRHQ): M 2 4 7

stock  foundation
XX K♦

tableau

```
    1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♠
 2  XX  XX  XX  6♥  6♣  Q♠
 3  K♥  Q♥  XX  5♥  5♣
 4           J♠  4♥  4♣
 5          10♠  3♥  3♣
 6           9♠  2♥  2♣
 7           8♠  A♥  A♣
 8           7♠
 9           6♠
10           5♠
11           4♠
12           3♠
13           2♠
14           A♠
```

Input an option (DMRHQ): M 3 4 6

stock  foundation
XX K♦  K♠

tableau

```
    1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣
 2  XX  XX  XX  6♥  6♣
 3  K♥  Q♥  J♣  5♥  5♣
 4               4♥  4♣
 5               3♥  3♣
 6               2♥  2♣
 7               A♥  A♣
```

Input an option (DMRHQ): M 1 3 6

```
stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  K♥
 2  K♣  XX  XX  6♥  6♣
 3      Q♥  J♣  5♥  5♣
 4              4♥  4♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 1 2 7

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1  Q♣  XX  XX  7♥  7♣  K♥  K♣
 2      XX  XX  6♥  6♣
 3      Q♥  J♣  5♥  5♣
 4              4♥  4♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 1 1 7

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1      XX  XX  7♥  7♣  K♥  K♣
 2      XX  XX  6♥  6♣      Q♣
 3      Q♥  J♣  5♥  5♣
 4              4♥  4♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 2 3 6

stock   foundation
 XX K♦  K♠
```

```
tableau
     1   2   3   4   5   6   7
 1      XX  XX  7♥  7♣  K♥  K♣
 2      J♥  XX  6♥  6♣  Q♥  Q♣
 3          J♣  5♥  5♣
 4              4♥  4♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 2 2 6

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1     10♥  XX  7♥  7♣  K♥  K♣
 2          XX  6♥  6♣  Q♥  Q♣
 3          J♣  5♥  5♣  J♥
 4              4♥  4♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 2 1 6

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1          XX  7♥  7♣  K♥  K♣
 2          XX  6♥  6♣  Q♥  Q♣
 3          J♣  5♥  5♣  J♥
 4              4♥  4♣ 10♥
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 3 3 7

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1          XX  7♥  7♣  K♥  K♣
 2         10♣  6♥  6♣  Q♥  Q♣
```

```
 3              5♥  5♣  J♥  J♣
 4              4♥  4♣ 10♥
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 3 2 7

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1          9♣  7♥  7♣  K♥  K♣
 2              6♥  6♣  Q♥  Q♣
 3              5♥  5♣  J♥  J♣
 4              4♥  4♣ 10♥ 10♣
 5              3♥  3♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): M 3 1 7

stock   foundation
 XX K♦  K♠

tableau
     1   2   3   4   5   6   7
 1              7♥  7♣  K♥  K♣
 2              6♥  6♣  Q♥  Q♣
 3              5♥  5♣  J♥  J♣
 4              4♥  4♣ 10♥ 10♣
 5              3♥  3♣      9♣
 6              2♥  2♣
 7              A♥  A♣


Input an option (DMRHQ): D

stock   foundation
   K♦  K♠

tableau
     1   2   3   4   5   6   7
 1  8♣  9♥  8♥  7♥  7♣  K♥  K♣
 2              6♥  6♣  Q♥  Q♣
 3              5♥  5♣  J♥  J♣
 4              4♥  4♣ 10♥ 10♣
 5              3♥  3♣      9♣
 6              2♥  2♣
```

```
 7              A♥  A♣


Input an option (DMRHQ): M 1 1 7

stock   foundation
   K♦  K♠

tableau
     1   2   3   4   5   6   7
 1       9♥  8♥  7♥  7♣  K♥  K♣
 2               6♥  6♣  Q♥  Q♣
 3               5♥  5♣  J♥  J♣
 4               4♥  4♣ 10♥ 10♣
 5               3♥  3♣      9♣
 6               2♥  2♣      8♣
 7               A♥  A♣


Input an option (DMRHQ): M 2 1 6

stock   foundation
   K♦  K♠

tableau
     1   2   3   4   5   6   7
 1           8♥  7♥  7♣  K♥  K♣
 2               6♥  6♣  Q♥  Q♣
 3               5♥  5♣  J♥  J♣
 4               4♥  4♣ 10♥ 10♣
 5               3♥  3♣  9♥  9♣
 6               2♥  2♣      8♣
 7               A♥  A♣


Input an option (DMRHQ): M 3 1 6

stock   foundation
   K♦  K♠

tableau
     1   2   3   4   5   6   7
 1               7♥  7♣  K♥  K♣
 2               6♥  6♣  Q♥  Q♣
 3               5♥  5♣  J♥  J♣
 4               4♥  4♣ 10♥ 10♣
 5               3♥  3♣  9♥  9♣
 6               2♥  2♣  8♥  8♣
 7               A♥  A♣


Input an option (DMRHQ): M 5 1 7
```

```
stock   foundation
   K♦  K♠  K♣

tableau
     1   2   3   4   5   6   7
 1                7♥      K♥
 2                6♥      Q♥
 3                5♥      J♥
 4                4♥     10♥
 5                3♥      9♥
 6                2♥      8♥
 7                A♥


Input an option (DMRHQ): M 4 1 6
You won!

New Game.

stock   foundation
 XX

tableau
     1   2   3   4   5   6   7
 1  XX  XX  XX  7♥  7♣  7♠  7♦
 2  XX  XX  XX  6♥  6♣  6♠  6♦
 3  XX  XX  XX  5♥  5♣  5♠  5♦
 4  Q♠  J♦  J♠  4♥  4♣  4♠  4♦
 5  Q♦ 10♦ 10♠  3♥  3♣  3♠  3♦
 6  K♦  9♦  9♠  2♥  2♣  2♠  2♦
 7  K♠  8♦  8♠  A♥  A♣  A♠  A♦


Input options:
    D: Deal to the Tableau (one card to first three columns).
    M c r d: Move card from Tableau (column,row) to end of column d.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
    Q: Quit the game


Input an option (DMRHQ): Q
Thank you for playing.
```

**Grading Rubric**

Computer Project #10
Scoring Summary

General Requirements:
(4 pts)  Coding Standard 1-9
(descriptive comments, mnemonic identifiers, format, etc…)

Implementations:

(6 pts) initialize
(3 pts) deal_from_stock
(8 pts) validate_move
(6 pts) move
(5 pts) check_sequence
(5 pts) move_to_foundation
(4 pts) check_for_win
(7 pts) Test 1
(7 pts) Test 2