CSE 231 **Spring 2016**

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 Monday, April 18.

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 **handin system** before the project deadline.

Assignment Background

Alaska is a popular 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/

Under the "Solitaire" tab, click on "Select Game…" and "Alaska".

Your program will allow the user to play Alaska with the program managing the game. The game rules are given below.

Game Rules

1. Start: The game is played with one standard deck of 52 cards.

All 52 cards are then dealt into a seven-column *tableau*. The first column has one card, face up; the other six columns each have five cards face up but with an increasing number of cards face down starting with one card face down in column 2 up to six cards face down in column 7. During play cards will be moved among columns.

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

2. Goal: The game is won when all cards have been moved from the tableau to the foundation.

3. Moves: A player can move a pile of one or more cards from the bottom of one tableau column to the bottom of another column.

A move is valid if the card at the "beginning" of the pile is the same suit as the destination card and their ranks differ by exactly one (greater than or less than).

If a column of the tableau becomes empty, only a pile with a King at the "beginning" can be moved to that column.

A card can be moved from the bottom of a tableau column to a foundation stack if the suits match and the tableau card's rank is exactly one greater than the top foundation card's rank. If a foundation stack is empty, only an Ace can be moved there.

After a move, if the card remaining at the bottom of a tableau column is face down, it will automatically be turned face up.

No other moves are permitted.

Assignment Specifications

You will develop a program that allows the user to play Alaska 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 below. We provide a program **`proj10.py`** that contains stubs of the required function as well as a driver. The provided program runs as is, but doesn't do anything useful. You are encouraged to create additional functions as needed. Your program must use the import statement for **`cards.py`** —you are not allowed to copy that file into your program. (Your program will be run with our copy of **`cards.py)`.**

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

`F x y`	Move card from Tableau column **`x`** to Foundation column **`y`**.
`T x y c`	Move c >= 1 cards from Tableau column **`x`** to Tableau column **`y`**.
`R`	Restart the game (after shuffling)
`H`	Display the menu of choices
`Q`	Quit

where **`x`** and **`y`** denote column numbers; Tableau columns are numbered from 1 to 7 and Foundation stacks are numbered from 1 to 4.

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`**" (or "**`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:

`init_game() → (tableau, foundation)`

That function has no parameters. It creates and initializes the tableau, and foundation, and then returns them as a tuple, in that order.

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

`display( tableau, foundation ) → None`

That function has two parameters: the data structure representing the tableau, and the data structure representing the foundation.

The foundation is displayed first. A non-empty foundation stack will be displayed as the top card in the stack (i.e. last card moved to it); and an empty stack will be displayed as whitespace.
The tableau is displayed second, each column of the tableau will be displayed downwards as shown in the sample below. An empty column will be displayed by whitespace.

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

`valid_move( card1, card2 ) → Bool`

That function has two parameters: the two cards to be compared. The function will return **`True`**, if the move is valid, i.e. if suits are the same and ranks differ by 1; and **`False`**, otherwise.

5. The program will use the following function to move a card from the tableau to the foundation:

`foundation_move( tableau, foundation, from_col, to_stack ) → Bool`

That function has four parameters: the data structure representing the tableau, the data structure representing the foundation, an **`int`** indicating the tableau column whose bottom card should be moved, and an **`int`** indicating the destination stack within the foundation.. If the move is valid, the function will update the tableau and foundation and return **`True`**; otherwise, it will do nothing to them and return **`False`**. (Note that you are not required to make use of the returned Boolean—it is only there if your design finds the value useful.)

6. The program will use the following function to move within the tableau:

`tableau_move ( tableau, from_col, to_col, count) → Bool`

That function has four parameters: the data structure representing the tableau, an **`int`** indicating the source column, and an **`int`** indicating the destination column, and an **`int`** indicating the size (>=1) of the pile being moved. If the move is valid, the function will update the tableau; otherwise, it will do

nothing to it. The function will return **`True`**, if the move is valid; and **`False`**, otherwise. (Note that you are not required to make use of the returned Boolean—it is only there if your design finds the value useful.)

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

`win( tableau, foundation ) → Bool`

That function has two parameters: the data structure representing the tableau and the data structure representing the foundation. It returns **`True`**, if all cards have been moved from the tableau to the foundation; and **`False`**, otherwise. (To determine the winner you may use the tableau or foundation or both—your choice.)

8. Mandatory error checking.
A. "Error: Invalid Move": invalid moves are not to be allowed. They generate an error message and cannot allow the state of the game to change. Details of *why* the move is invalid are not required to be part of the error message (but you can do so if you wish).
B. "Error: Incorrect Command": an incorrect command cannot allow the state of the game to change, and will generate an error message. You *must* also indicate in your error message these types of error:
I. "Incorrect Command", i.e. a command not in the menu.
II. "Incorrect number of arguments"
III. "Incorrect type of arguments", e.g. specifying a letter when an integer is expected
If multiple errors apply, printing any one of the errors is fine, e.g. if the number of arguments and the types of the arguments are both wrong, an error message about either is fine (but if you want to print information about both errors, that is OK, but not required).

Assignment Notes

1. Before you begin to write any code, play with the game at http://worldofsolitaire.com/ and look over the sample interaction below to be sure you understand the rules of the game and how you will simulate the game. The demo program is at http://worldofsolitaire.com/: Under the "Solitaire" tab, click on "Select Game…" and "Alaska".

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!*

3. Laboratory Exercise #12 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. It runs as is, but does nothing useful. Gradually replace the "stub" code (marked with comments) with your own code. (Delete the stub code.)

5. Displaying the tableau in columns is tricky. Start by implementing a very simple **`display`** function: You can easily label and display the foundation on one line, and the tableau using seven lines, with each line displaying the cards in a column. Once you have the game logic working

properly, modify your display function to display the current game state as described in the specification for function **`display`**. (Displaying the tableau in columns instead of rows will cost only a small number of points compared to enforcing the rules of the game.)

6. Python has an absolute value function, `abs()`. You likely will find it useful in this project.

7. 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.

8. 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.

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

SAMPLE OUTPUT

```
Alaska Card Game:
     Foundation: Columns are numbered 1, 2, 3, 4
                 Built up by rank and by suit from Ace to King.
                 The top card may be moved.
     Tableau: Columns are numbered 1,2,3,4,5,6,7
              Built up or down by rank and by suit.
              The top card may be moved.
              Complete or partial face-up piles may be moved.
              An empty spot may be filled with a King or a pile starting with a
King.
     To win, all cards must be in the Foundation.
============================

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

Input options:
    F x y : Move card from Tableau column x to Foundation y.
    T x y c: Move pile of length c >= 1 from Tableau column x to Tableau column y.
    R: Restart the game (after shuffling)
    H: Display the menu of choices
```

```
    Q: Quit the game


Enter a choice: F 1 1
============================
 A♥
----------------------------
     XX  XX  XX  XX  XX  XX
    10♠  XX  XX  XX  XX  XX
     7♦  6♦  XX  XX  XX  XX
     K♠ 10♥  9♦  XX  XX  XX
    10♣  6♠  Q♠  J♣  XX  XX
     4♥  9♣  7♥  4♣  9♥  XX
         3♥  5♣  2♦  A♠  3♣
             Q♥  A♣  Q♦  6♥
                 4♦  2♠  5♦
                     8♥  4♠
                         3♠

Enter a choice: T 2 1 3
============================
 A♥
----------------------------
 K♠  XX  XX  XX  XX  XX  XX
10♣ 10♠  XX  XX  XX  XX  XX
 4♥  7♦  6♦  XX  XX  XX  XX
        10♥  9♦  XX  XX  XX
         6♠  Q♠  J♣  XX  XX
         9♣  7♥  4♣  9♥  XX
         3♥  5♣  2♦  A♠  3♣
             Q♥  A♣  Q♦  6♥
                 4♦  2♠  5♦
                     8♥  4♠
                         3♠

Enter a choice: T 3 2 5
============================
 A♥
----------------------------
 K♠  XX  XX  XX  XX  XX  XX
10♣ 10♠  J♥  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  A♠  3♣
     3♥      Q♥  A♣  Q♦  6♥
                 4♦  2♠  5♦
                     8♥  4♠
                         3♠

Enter a choice: T 6 7 2
============================
 A♥
----------------------------
 K♠  XX  XX  XX  XX  XX  XX
10♣ 10♠  J♥  XX  XX  XX  XX
```

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

Enter a choice: T 3 4 1
============================
 A♥
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠      XX  XX  XX  XX
 4♥  7♦       XX  XX  XX  XX
     6♦       9♦  XX  XX  XX
    10♥       Q♠  J♣  XX  XX
     6♠       7♥  4♣  9♥  XX
     9♣       5♣  2♦  A♠  3♣
     3♥       Q♥  A♣  Q♦  6♥
              J♥  4♦      5♦
                          4♠
                          3♠
                          2♠
                          8♥

Enter a choice: T 6 3 1
============================
 A♥
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦       XX  XX  XX  XX
     6♦       9♦  XX  XX  XX
    10♥       Q♠  J♣  XX  XX
     6♠       7♥  4♣  9♥  XX
     9♣       5♣  2♦  A♠  3♣
     3♥       Q♥  A♣      6♥
              J♥  4♦      5♦
                          4♠
                          3♠
                          2♠
                          8♥

Enter a choice: F 6 2
============================
 A♥  A♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦       XX  XX  XX  XX
     6♦       9♦  XX  XX  XX
    10♥       Q♠  J♣  XX  XX
```

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

Enter a choice: T 7 6 1
============================
 A♥  A♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥  A♣      6♥
             J♥  4♦      5♦
                         4♠
                         3♠
                         2♠

Enter a choice: F 7 2
============================
 A♥  2♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥  A♣      6♥
             J♥  4♦      5♦
                         4♠
                         3♠

Enter a choice: F 7 2
============================
 A♥  3♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥  A♣      6♥
             J♥  4♦      5♦
                         4♠
```

```
Enter a choice: F 7 2
============================
 A♥  4♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥  A♣      6♥
             J♥  4♦      5♦

Enter a choice: T 5 7 1
============================
 A♥  4♠
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥  A♣      6♥
             J♥          5♦
                         4♦

Enter a choice: F 5 3
============================
 A♥  4♠  A♣
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
     6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
     3♥      Q♥          6♥
             J♥          5♦
                         4♦

Enter a choice: T 2 1 1
============================
 A♥  4♠  A♣
----------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
 3♥  6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠      7♥  4♣  9♥  XX
     9♣      5♣  2♦  8♥  3♣
             Q♥          6♥
             J♥          5♦
```

```
                          4♦

Enter a choice: T 4 6 4
============================
 A♥  4♠  A♣
--------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
 3♥  6♦      9♦  XX  XX  XX
    10♥      Q♠  J♣  XX  XX
     6♠          4♣  9♥  XX
     9♣          2♦  8♥  3♣
                     7♥  6♥
                     5♣  5♦
                     Q♥  4♦
                     J♥

Enter a choice: T 2 6 3
============================
 A♥  4♠  A♣
--------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
 3♥  6♦      9♦  XX  XX  XX
             Q♠  J♣  XX  XX
                 4♣  9♥  XX
                 2♦  8♥  3♣
                     7♥  6♥
                     5♣  5♦
                     Q♥  4♦
                     J♥
                    10♥
                     6♠
                     9♣

Enter a choice: T 7 2 2
============================
 A♥  4♠  A♣
--------------------------
 K♠  XX  K♦  XX  XX  XX  XX
10♣ 10♠  Q♦  XX  XX  XX  XX
 4♥  7♦      XX  XX  XX  XX
 3♥  6♦      9♦  XX  XX  XX
     5♦      Q♠  J♣  XX  XX
     4♦          4♣  9♥  XX
                 2♦  8♥  3♣
                     7♥  6♥
                     5♣
                     Q♥
                     J♥
                    10♥
                     6♠
                     9♣

Enter a choice: q
```