ENGR 160 Algorithms and Applications Sec 02 Spring 2018

Homework Assignment #9 Due 9:00am Mar 16 (F) 2018

Read Chapter 7 of the MATLAB book.

Problem 9.1

Create and run a Matlab program plot the following function within a user specified range, with the user specified color and marker style. Your program should prompt the user to enter a valid positive number n that is greater than 2 (n>2), a valid color (‘red’, ‘green’ or ‘blue’), and a valid data marker style (‘*’ or ‘o’). Perform error-checking for all user’s inputs, and plot the following function within the range of $0 \le x \le n$, with the specified color and style.

$$y = \begin{cases} e^{-x/2}, & 0 <= x <= 2; \\ e^{-1}(x-1)^2, & 2 <= x <= n; \end{cases}$$

The title of the plot should be “y = f(x) with x in the range of 0 and n”, where n should be replaced by the actual value of n inputted by the user. Test your program as follows:

```
Please enter the upper bound of x: 2
Invalid value! The upper bound should be a value greater than 2
Please enter the upper bound of x: 4
Enter a color; red, green, or blue: yellow
Invalid color!
Enter a color; red, green, or blue: red
Enter a plot style; * or o: -
Invalid style!
Enter a plot style; * or o: *
```

Attach your testing result and figure.

Problem 9.2

Write a Matlab function called **ispalindrome** to accept a string as input argument and determine if this string is a palindrome. The function should return a logical TRUE value if it is and a logical FALSE if it is not and output appropriate messages. A palindrome is a sentence that reads the same backward as forward, such as “Reward a Toyota drawer” or Napoleon’s classic lament, “Able was I ere I saw Elba”. Punctuation marks, blanks, and any other non-letter characters shall have no effects. The decision is not case-sensitive. Test your function with the two sentences given above, then “Today is such a nice day!” and “A man, a plan, a cat, a canal Panama!”. Your last test should output the following message:

'A man, a plan, a cat, a canal Panama!' is a palindrome!

Problem 9.3

Create and run a Matlab program called **wordscramble.m** to randomly scramble a given word for N times. The word to be scrambled and N should be entered by the user input. Error-check the input N to make sure it is a positive integer. As the result of running your program, there should be N variables, each with a name as ‘word1’, ‘word2’, … , storing the N scrambled words, respectively. Use disp function to display the original word as well as all the scrambled words as follows:

```
Please enter a word: example
How many times do you want to scramble the word?  5
word = example
```

```
word1 = leapmxe
word2 = pexamle
word3 = lapxeem
word4 = mpxeale
word5 = lexamep
```

(hint: you can consider using the MATLAB built-in function ***randperm***.)

Problem 9.4

Create and run a MATLAB program that "shuffles" a deck of 52 playing cards. Your program should include the following files.

1. A main script file named **cards.m**, which calls the following functions.
2. A function named **new_deck**, which generates a new deck of 52 cards, stored and returned in a string matrix with 52 rows, each representing one card. The new deck should be sorted by both rank (from A to Q) and suit (in the order of club, diamond, heart, spades).
3. A function named **shuffle_cards**, which takes a string matrix representing any number of cards in any rank or suit order as input, randomly reorder them, and return the shuffled cards in a new string matrix with the new order.
4. A function named **disp_cards**, which displays all cards given in a string matrix. For example, when displaying a new deck of cards, it should print out the following lines:

```
A  club
A  diamond
A  heart
A  spade
2  club
2  diamond
2  heart
2  spade
…
K  club
K  diamond
K  heart
K  spade
```

After shuffling, when display the shuffled cards, the lines will be reordered. An example of the first 7 out of 52 rows in the output is shown.

```
2 spades
Q hearts
8 spades
A diamonds
6 clubs
10 hearts
J clubs
…
```

You should be able to modify your main script only if you want to reshuffle the same cards multiple times by calling the **shuffle_cards** function multiple times. You should also be able to modify your main script only if you want to generate more than one deck of cards, by calling **new_deck** multiple times. Show the results of one new deck generated by your program, one deck of cards after being shuffled once, as well as two deck of cards being shuffled twice.