% PlotFromUser.m
% Patrick Utz, 3/16/18, 9.1

% Problem: 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 ? x ? 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:

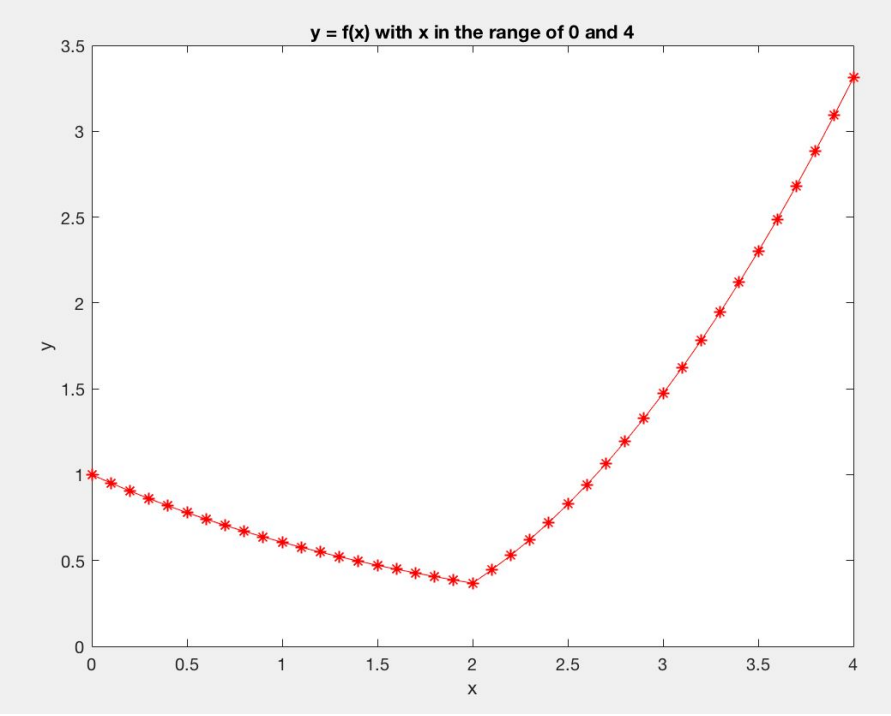
Please enter the upper bound of x: 2
Invalid value! The upper bound should be a value greater than 2
Please enter the upper bound of x: 4
Enter a color; red, green, or blue: yellow
Invalid color!
Enter a color; red, green, or blue: red
Enter a plot style; * or o: -
Invalid style!
Enter a plot style;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

% Variables: upperBound = upperBound input; color/colorNoBlank = inputted
% color; marker/markerNoBlank = inputted marker; used standard variables
% for plot

```
clear
upperBound = input('Please enter the upper bound of x that is greater than 2: ');
while upperBound <= 2
   fprintf('\nInvalid value! The upper bound should be a value greater than 2.\n');
   upperBound = input('Please enter the upper bound of x that is greater than 2: ');
end
color = input('\nEnter a color; red, green, or blue: ', 's');
colorNoBlank = strtrim(color);
while ~(strcmp(colorNoBlank,'red')) && ~(strcmp(colorNoBlank,'green')) &&
~(strcmp(colorNoBlank,'blue'))
   fprintf('Invalid color!\n');
   color = input('\nEnter a color; red, green, or blue: ', 's');
   colorNoBlank = strtrim(color);
end
marker = input('\nEnter a plot style; * or o: ', 's');
markerNoBlank = strtrim(marker);
```

```
while ~(strcmp(markerNoBlank,'*')) && ~(strcmp(markerNoBlank,'o'))
    fprintf('Invalid style!\n');
    marker = input('\nEnter a plot style; * or o: ', 's');
    markerNoBlank = strtrim(marker);
end

x = 0:0.1:upperBound;
y = (exp(-x./2)).*(x >= 0 & x <= 2) + (exp(-1).*(x-1).^2).*(x > 2 & x <= upperBound);
plot(x,y,colorNoBlank,'Marker',markerNoBlank);
customTitle = sprintf('y = f(x) with x in the range of 0 and %s',num2str(upperBound));
title(customTitle);
xlabel('x');
ylabel('y');
```

```
>> PlotFromUser
Please enter the upper bound of x that is greater than 2: 2

Invalid value! The upper bound should be a value greater than 2.
Please enter the upper bound of x that is greater than 2: 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: *
>>
```



```
% ispalindrome.m
% Patrick Utz, 3/16/18, 9.2

% Problem: 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!

% Variables: onlyLetterString = turns input into pure letters;
% testString = makes all letters lower case and ready to test;
% flipString = flips the testString left to right

function result = ispalindrome(stringIn)
% ispalindrome calculates if input string is a palindrome
% Format of call: ispalindrome( input string )
% Returns a logical TRUE value if it is a palindrome and a logical FALSE
% if it is not

onlyLetterString = stringIn(isletter(stringIn));
testString = lower(onlyLetterString);
flipString = fliplr(testString);
if testString == flipString
   result = logical(1);
else
   result = logical(0);
end
end
```

% test program for problem two

```
testString = 'A man, a plan, a cat, a canal Panama!';
if ispalindrome(testString)
   fprintf('"%s" is a palindrome!\n',testString)
else
   fprintf('"%s" is not a palindrome!\n',testString)
end
```

```
>> isPalindromeTest
'Reward a Toyota drawer' is a palindrome!
>> isPalindromeTest
'Able was I ere I saw Elba' is a palindrome!
>> isPalindromeTest
'Today is such a nice day!' is not a palindrome!
>> isPalindromeTest
'A man, a plan, a cat, a canal Panama!' is a palindrome!
>>
```

% wordscramble.m
% Patrick Utz, 3/16/18, 9.3

% Problem: 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:

% Variables: word = inputted word; n = amount of times that user wants
% the word to be scrambled; result = final result; random = refernce array
% for indices of the scramble; scarmble = scrambled words; number = temp
% conversion of the number n to be displayed

```
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***.)

```
clear
word = input('Please enter a word: ', 's');
n = input('How many times do you want to scramble the word? ');
while n < 1 || int32(n) ~= n
    fprintf('Invalid input.\n')
    n = input('How many times do you want to scramble the word? ');
end
result = ['word  = ', 'example'];
for k = 1:n
    random = randperm(length(word));
    scramble = [''];
    for m = 1:length(word)
        scramble(m) = word(random(m));
    end
    number = num2str(k);
```

```
    result(k+1,:) = ['word' number ' = ' scramble];
end
disp(result)


>> wordscramble
Please enter a word: example
How many times do you want to scramble the word? 5
word  = example
word1 = pmlaxee
word2 = peexlma
word3 = lepxema
word4 = eplaemx
word5 = xmpeale
>>
```

% cards.m
% Patrick Utz, 3/16/18, 9.4

% Problem: 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.

% Variables: newDeck = a new deck of cards; shuffledOnce = a deck of
% cards that has been shuffled once; shuffledTwice = a deck of cards that
% has been shuffled twice

```
clear
fprintf('One New Deck: \n')
newDeck = new_deck();
disp_cards(newDeck)
fprintf('\n\n One Deck Shuffled Once: \n')
shuffledOnce = shuffle_cards(newDeck);
disp_cards(shuffledOnce)
fprintf('\n\n Two Decks Shuffled Twice: \n')
shuffledOnce1 = shuffle_cards(newDeck);
shuffledOnce2 = shuffle_cards(newDeck);
shuffledTwice1 = shuffle_cards(shuffledOnce1);
```

```
disp_cards(shuffledTwice1)
shuffledTwice2 = shuffle_cards(shuffledOnce2);
disp_cards(shuffledTwice2)
```

% new_deck.m
% Patrick Utz, 3/16/18, 9.4

```
function newDeck = new_deck()
% new_deck creates a new deck in a string matrix form
% Format of call: new_deck()
% Returns a string matrix representing a new deck of cards

newDeck = strings();
ranks = ["A" "2" "3" "4" "5" "6" "7" "8" "9" "10" "J" "Q" "K"];
suits = ["club" "diamond" "heart" "spade"];
n = 0;
allRanks = strings();
allSuits = strings();
for j = 1:13
    n = n+1;
    p = size(allRanks,1);
    for i = 1:4
        o = j*i;
        allRanks(p+i,1) = ranks(n);
    end
end

for j = 1:13
   p = size(allSuits,1);
   for i = 1:4
       o = j*i;
       allSuits(p+i,2) = suits(i);
   end
end

allRanks(1) = [];
allSuits(1,:) = [];

for d = 1:52
    allRanks(d,2) = allSuits(d,2);
end
newDeck = allRanks;
```

```
% shuffle_cards.m
% Patrick Utz, 3/16/18, 9.4

function shuffle = shuffle_cards(inputStr)
% shuffle_cards shuffles a given set of cards
% Format of call: shuffle_cards(string array representing deck to be shuffled)
% Returns a string matrix of the shuffled deck

random = randperm(length(inputStr));
shuffle = strings(1,2);

for k = 1:length(inputStr)
   shuffle(k,:) = inputStr(random(k),:);
end
```

```
% disp_cards.m
% Patrick Utz, 3/16/18, 9.4

function disp_cards(inputStr)
% disp_cards converts a string array of a card deck and displays it
% Format of call: disp_cards(string array of card deck want to display)
% Displays the deck of cards to the user

for k = 1:length(inputStr);
   fprintf('%s %s \n', inputStr(k,1), inputStr(k,2))
end
```

FORMATTED RESULT INTO 3 COLUMNS TO SAVE PAGE SPACE

```
>> cards
One New Deck:
A club
A diamond
A heart
A spade
2 club
2 diamond
2 heart
2 spade
3 club
3 diamond
3 heart
3 spade
4 club
4 diamond
4 heart
4 spade
5 club
5 diamond
5 heart
5 spade
6 club
6 diamond
6 heart
6 spade
7 club
7 diamond
7 heart
7 spade
8 club
8 diamond
8 heart
8 spade
9 club
9 diamond
9 heart
9 spade
10 club
10 diamond
10 heart
10 spade
J club
J diamond
J heart
J spade
Q club
Q diamond
Q heart
Q spade
K club
K diamond
K heart
K spade
```

One Deck Shuffled Once:

Q heart
10 diamond
2 club
9 club
8 club
2 diamond
10 club
A diamond
9 spade
5 club
3 heart
A spade
4 heart
Q spade
4 club
3 club
9 heart
K diamond
5 spade
2 spade
7 spade
4 spade
10 heart
8 diamond
K heart
7 club
6 heart
A heart
J diamond
7 diamond
J club
A club
J heart
J spade
8 spade
10 spade
5 diamond
K club
6 spade
3 diamond
8 heart
2 heart
K spade
5 heart
7 heart
Q club
Q diamond
4 diamond
3 spade
9 diamond
6 diamond
6 club

Two Decks Shuffled Twice:

4 diamond
6 diamond
9 heart
4 heart
5 heart
3 heart
Q heart
A club
3 spade
J club
Q spade
8 heart
4 club
2 spade
K club
9 diamond
4 spade
6 spade
2 heart
K diamond
7 club
9 club
7 heart
10 spade
5 diamond
J diamond
7 diamond
8 club
2 club
Q club
10 heart
J heart
8 diamond
K spade
A heart
5 spade
3 club
10 club
5 club
3 diamond
A diamond
K heart
8 spade
6 club
7 spade
10 diamond
Q diamond
6 heart
9 spade
A spade
2 diamond
J spade
K heart
J heart
3 heart
2 diamond
3 club
4 spade
Q diamond
K diamond
4 diamond
7 club
10 diamond
A spade
9 spade
5 heart
J diamond
7 diamond
Q heart
6 spade
6 diamond
6 heart
9 heart
4 club
8 club
5 club
4 heart
J club
3 spade
7 heart
2 spade
K spade
6 club
5 spade
10 spade
9 club

8 heart
A heart
A diamond
7 spade
J spade
8 diamond
Q spade

10 heart
A club
9 diamond
5 diamond
Q club
10 club
8 spade

2 heart
3 diamond
2 club
K club
>>