

Project
CSCE 4240 and CSCE 5225
Spring 2019

All work must be submitted through CANVAS via the course site by 11:59pm on or before the due date.

CSCE 4240

1. Implement **snake_manual_input** for which the specifications are as follows:

```
function [x, y] = snake_manual_input(f, np, style)
% SNAKE_MANUAL_INPUT Manual input of initial snake.
% [X, Y] = SNAKE_MANUAL_INPUT(F, NP, STYLE) generates the (x,y)
% coordinates on an np-point initial snake for a region of image
% F. The user inputs vertices manually and then this function
% generates points in between the vertices so that the snake has
% NP equally-spaced points. When the function is called, it
% outputs the instructions for inputting the vertices. The
% snake coordinates, [x, y] form a closed polygon.
%
% STYLE is a character string made from one element from any or
% all the following 3 columns:
%
%      b  blue      .  point      -  solid
%      g  green     o  circle     :  dotted
%      r  red       x  x-mark    -. dashdot
%      c  cyan      +  plus       -- dashed
%      m  magenta   *  star       (none) no line
%      y  yellow    s  square
%      k  black     d  diamond
%      w  white     v  triangle (down)
%                  ^  triangle (up)
%                  <  triangle (left)
%                  >  triangle (right)
%                  p  pentagram
%                  h  hexagram
```

```

%
%
% Thus, a symbol from the first column gives the color, the second
% is the symbol used in the plot, and the third specifies the type
% of line used to join the points in the plot. For example, to
% plot red circles joined by straight lines we specify the string
% 'ro-'. To plot just red circles without any connecting lines we
% specify the string 'ro'. The default MATLAB plot is black points
% joined by black solid lines. The default for the present
% function is black dots with no lines.

```

Add to the header the names of the team members and contact information. The code for this function is in at the web site provided by the textbook authors. You may use that code for reference but your solution should not be a direct copy.

2. Implement **snake_display(x, y, style)** for which the specifications are as follows:

```

function snake_display(x, y, style)
%SNAKE_DISPLAY 2D display of a snake curve,
% SNAKE_DISPLAY(X, Y, STYLE) displays the coordinates (x, y) of a
% snake. STYLE is a character string made from one element from
% any or all the following 3 columns:
%
%      b  blue      .  point      -  solid
%      g  green     o  circle     :  dotted
%      r  red       x  x-mark     -. dashdot
%      c  cyan      +  plus       -- dashed
%      m  magenta   *  star       (none) no line
%      y  yellow    s  square
%      k  black     d  diamond
%      w  white     v  triangle (down)
%                  ^  triangle (up)
%                  <  triangle (left)
%                  >  triangle (right)
%                  p  pentagram
%                  h  hexagram
%
%
% Thus, a symbol from the first column gives the color, the second
% the symbol used in the plot, and the third specified the type of
% line used to join the points in the plot. For example, to plot

```

% red circles joined by dotted lines we specify the string
% 'ro:'. To plot just red circles without any connecting lines we
% specify the string 'ro'. The default MATLAB plot is black points
% joined by black solid lines. The default for the present
% function is black dots with no lines.


Add to the header the names of the team members and contact information. The code for this function is in at the web site provided by the textbook authors. It consists of a single line of code because this function corresponds to a built-in Matlab function. Thus, if you use Matlab, your solution will be exactly like that of the authors. (That will be o.k.)

3. TBD. There will be three or four tests prescribed that demonstrate three or four style combinations using one TBD image.

Due Date: Feb. 1

Turn-in: Using as input the image rose512.tif (1) a “main” program that accepts the manual input of a snake contour and displays the contour as a solid line; (2) a subprogram, snake_manual_input; (3) a subprogram, snake_display; (4) three examples of execution – one with yellow circles, one with blue diamonds, and one with red squares. The contours do not necessarily need to be displayed upon the input image. That is, the contours may be displayed on a blank, i.e., “black” image.

Only one Turn-in per team is required. Put the names of the programmers in the header along with the team name (if any).



CSCE 5225

1. Read sections 11.1 and 11.2, stopping at “MOG.” (Eventually, we will read the rest of section 11.2.) In the fourth edition this is pages 877-888. The key equation in this part of the material eq. 11-46. (Note that the **F** will not be formally defined when you reach this point.) Some points to which you might pay attention are:
 - a. Which variables express properties of the contour points and which variables express properties computed over the entire image?
 - b. What is the dimensionality of each variable? [This is related to a.) above.]
2. Read projects 11.1 and 11.2. In the fourth edition this is pages 947-948.

Due Date: None

Turn-in: None