Joanne Kwon

PID: A15359545

**Part I**

**QUESTION #1**

* I have replaced both remaining ??? with a value of 240. This is because, in order for the background area to be white we need to make all three layers of trekkie.png to have the same values. Usually, these values are high for the color white.

**QUESTION #2**

Code:

>> trekkie = imread ('trekkie.png');

>> tiedye = imread ('tiedye.png');

>> space = imread ('space.png');

>> filter = trekkie (:,:,1) > 240 & trekkie (:,:,2)>240 & trekkie (:,:,3)>240;

>> filter = cat(3,filter,filter,filter);

>> mashup = trekkie;

>> mashup(filter) = space(filter);

>> imshow(mashup);

>> filter = trekkie(:,:,1) > 100 & trekkie(:,:,2) < 50 & trekkie(:,:,3) < 50;

>> filter = cat(3,filter,filter,filter);

>> moremashup = mashup;

>> moremashup(filter) = tiedye(filter);

>> imshow(moremashup);

* I used the conditionals >225, >255, and >255 for ‘mashup’ (trekki and space) because if the red, green, and blue layers of the photo are set to 255 it creates white, which is what we needed for a ‘white screen’. I used the conditionals >100, <50, and <50 for ‘moremashup’ (mashup and tiedye) because it showed the best results when trying to replace the red with tiedye.

**QUESTION #3**



**QUESTION #4**

* If we set ‘x=0’ and then evaluate the sums, there is an error because the value 0 is not a real positive integer. Subscript indices have to be either a real positive integer or logicals. If we set ‘x=5’ and then evaluate the sums, there is an error because the index exceeds matrix dimensions.

**QUESTION #5**

* The version of this conditional code that results in an error is ‘(x>=1 & x<=4) & sum(A(x,:)) == 34’ (the code with just one &).

**Part II**

**IsValidMove**

Code:

function [trueORfalse] = IsValidMove(x, y)

% Name: Joanne Kwon

% The function 'IsValidMove' takes two arguments (x and y) and determines if

% the values make sense. In other words, 'IsValidFunction' testes if x and y

% take one values from 1 to 4.

% The argument x is the row and argument y is the colum of the tile we

% would like to slide into the empty tile position. After the arguments

% are taken, 'IsValidMove' should return true (1) if x and y are valid

% indices for the board matrix and should return false (0) otherwise.

if (x >= 1 && x <= 4 && y >= 1 && y <= 4)

trueORfalse = 1;

else

trueORfalse = 0;

end

**MoveTile**

Code:

function [newmatrix] = MoveTile(board, x, y)

% Name: Joanne Kwon

% The Function 'MoveTile' takes three arguments (board, x, and y) and

% returns an updated board state with the appropriate tile swapping

% position with the empty tile.

% The argument board is a 4x4 matrix that represents the position of the

% tiles with 16 representing the empty tile. The argument x is the row

% and y is the column of the tile we would like the slide into the empty

% tile position.

if IsValidMove (x, y) == 1

if (IsValidMove(x, y-1) == 1 & board(x, y-1) == 16)

board(x, y-1) = board(x, y);

board (x,y) = 16;

elseif (IsValidMove(x-1, y) == 1 & board(x-1, y) == 16)

board (x-1, y) = board (x, y);

board (x, y) = 16;

elseif (IsValidMove(x, y+1) == 1 & board(x, y+1) == 16)

board (x, y+1) = board (x, y);

board (x, y) = 16;

elseif (IsValidMove(x+1, y) == 1 & board(x+1, y) == 16)

board (x+1, y) = board (x,y);

board(x, y) = 16;

else

board();

end

else

board();

end

newmatrix = board;