CS242
JavaScript Lab Practice Questions

1. According to Wikipedia a happy number is defined by the following process : "Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers (or sad numbers)". Write a Javascript program that accepts a number from user and alerts whether the input number is happy or not.
   Some examples are : 1, 7, 10, 13, 19, 23
2. Write a Javascript function to explore string functions for the following:
   a. to check whether an `input` is a string or not.
   b. to check whether a string is blank or not.
   c. to split a string and convert it into an array of words.
   d. to validate an email address.
      [Note: Use pop-up box]
3. Write a simple JavaScript program to accept sequence of comma-separated integers from user. Convert the sequence into an array of numbers and sort the sequence in ascending order.
4. Write a simple JavaScript program to join all elements of the following array into a string.
5. You'll create a simple word guessing game where the user gets infinite tries to guess the word. Follow,
   a. Create two global arrays: one to hold the letters of the word (e.g. 'F', 'O', 'X'), and one to hold the current guessed letters (e.g. it would start with '_', '_', '_' and end with 'F', 'O', 'X').
   b. Write a function called guessLetter that will:
      i. Take one argument, the guessed letter.

    ii. Iterate through the word letters and see if the guessed letter is in there.
    iii. If the guessed letter matches a word letter, change the guessed letters array to reflect that.
    iv. When it's done iterating, it should log the current guessed letters ('F__')
    v. and congratulate the user if they found a new letter.
    vi. It should also figure out if there are any more letters that need to be guessed,
    vii. and if not, it should congratulate the user for winning the game.
6. Take the previous problem and make it Hangman,
    a. Keep track of all the guessed letters (right and wrong) and only let the user guess a letter once. If they guess a letter twice, do nothing.
    b. Keep track of the state of the hangman as a number (starting at 0), and subtract or add to that number every time they make a wrong guess.
    c. Once the number reaches 6 (a reasonable number of body parts for a hangman), inform the user that they lost.