CS1428 Lab 11h

# Name: Section:

1. (100pts) Write a program that calculates what is known as a Happy Prime number. The program should take the input of a single number and tell you if the number is a Happy Prime, a Happy non-prime, a Sad Prime, or a Sad non-prime.

**Happy Numbers:**

Starting with any [positive](http://en.wikipedia.org/wiki/Positive_number) [integer](http://en.wikipedia.org/wiki/Integer), replace the number by the [sum](http://en.wikipedia.org/wiki/Sum) of the squares of its [digits](http://en.wikipedia.org/wiki/Numerical_digit), 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).

For example, 19 is happy, as the associated sequence is:

12 + 92 = 82

82 + 22 = 68

62 + 82 = 100

12 + 02 + 02 = 1.

For example, 5 is unhappy(sad), as the associated sequence is:

52 = 25

22 + 52 = 29

22 + 92 = 85

82 + 52 = 89

82 + 92 = 145

12 + 42 + 52 = 42

42 + 22 = 20

22 + 02 = 4

42 = 16

12 + 62 = 37

32 + 72 = 30

32 + 02 = 9

92 = 81

82 + 12 = 65

62 + 52 = 61

62 + 12 = 37

This program will use your knowledge of sorting and searching. The program should use functions for all main forms of processing.

The algorithm to find a happy number works as such: You will first need to add the squares of the digits of the number together. You need to keep track of all the iterations of sums to be able to detect if a loop has started. The array, named cycle, can be 20 elements long as long as you don’t exceed the maximum value of an integer.

Steps:

* 1. Get Sum
  2. Try to add number to cycle
     1. Check to see if that number already exists, if so return false. Else add number, sort array, return true;

You will need 5 functions

**bool isPrime(…)** This function will check to see if the number is Prime. (Feel free to use your previous lab code for this)

**bool isHappy(…)** This function will check to see if the number is Happy.

**bool cycleAdd(…)** This function attempts to add the latest sum to the array(cycle). If the number already exists within the array, the function returns false. If the number doesn’t exist in the array it adds it, sorts the array, and returns true.

**bool cycleContains(…)** Checks to see if an number is in the array using a linear search.

**void selSort(…)** Sorts the array using the selection sort algorithm.

**Bonus:** Implement a binary search instead of a linear search.