15110 Summer 2018
Problem Set 6

Name: _____

Andrew ID: _____

Instructions
- Type or neatly write the answers to the following problems.
- Save or scan this file as a pdf and submit to Gradescope

Exercises

1. (1 pt) The slides 18-21 (can be a few slides back or forward) from Lecture Notes (Arrays) show how to compute the total sum of all integers in a two-dimensional list as follows:

```
def print_sum(matrix):
    sum = 0
    for row in range(0,len(matrix)):
        for col in range(0,len(matrix[row])):
            sum = sum + matrix[row][col]
    print(sum)
```

Show how to modify this function so that it prints the sum of each row of the two-dimensional list. (HINT: Two of the statements will have to move to different places in the function.)

2. (2 pts) A music playing application stores a sequence of four-minute music files for playback. (All songs are the same length of time.) The music files can be stored in one of two ways:

Algorithm 1: Songs are stored in the computer's memory contiguously, in the order of playback starting at a specific fixed location in computer memory which cannot be changed.

Algorithm 2: Songs are stored in the computer's memory in arbitrary order. Each song has a code that indicates the location in memory of the song that plays next.

The player keeps track of only the location of the first song in the playback sequence.

a. For each of the algorithms above, state which data structure appears to be in use? Array or a linked list?

b. If the application user wants to skip to the 50th song in the playback list, which algorithm will allow the user to hear this song faster? Explain your answer by describing what each algorithm would do to accomplish this task.

c. If the application user wants to insert a new song at the beginning of the playlist, which algorithm will allow the user to complete this operation faster? Explain your answer by describing what each algorithm would do to accomplish this task.

3. (2 pts) A hash table has 13 buckets (i.e. its table size is 13). When keys are stored in the hash table, we use the following hash function:

```
def h(word, table_size):
    k = 0
    for i in range(0,len(word)):
        k = ord(word[i]) + k * 128
    return k % table_size
```

In the function above, `ord(word[i])` returns the ASCII code of the ith character of `word`. Here are the ASCII codes for the lowercase letters:

a	b	c	d	e	f	g	h	i	j	k	l	m
97	98	99	100	101	102	103	104	105	106	107	108	109

n	o	p	q	r	s	t	u	v	w	x	y	z
110	111	112	113	114	115	116	117	118	119	120	121	122

a. Given the hash function above, in which bucket would the following words be stored? Show all steps of the computation to show the bucket that is chosen. Do not write the final answer only.
   - how

   - now

   - ban

   - cow

b. Do any of these words collide if we were to put them into a hash table of size 13 using the hash function above? Why or why not?

c. If 3n words are put into a hash table with n buckets so that the number of words per bucket is 3, what is the order of complexity to search for a word in the hash table in big O notation? Briefly explain your answer.

4. (3 pts) This problem deals with a binary tree known as a binary search tree.
   a. Insert the following integers into a binary search tree one at a time in the order given. Show the tree at each insertion step. You will draw 10 trees where the final step shows the resulting binary search tree.

   39 23 50 77 98 34 19 45 42 10

   b. Suppose you are now looking for the key 42 in your binary search tree. Which keys in the tree do you examine during your search? List the keys you have to examine in the order you examine them.

c. Draw another binary search tree that has the same set of keys organized in a different way. Hint: You can obtain such a tree by considering a case where the keys above are inserted in a different order.

d. What is the worst-case order of complexity for binary search provided that the binary search tree is balanced? What is it in the general case where we cannot assume anything about the shape of the tree?

e. Suppose that information about members of a club is stored using a binary search tree. The club inserts the information about each member into the data set using their age as the key upon joining. Considering the efficiency of the operation of searching for a given member using their age, what would constitute a worst-case scenario for this club? Your answer should involve a particular order in which the members join the club.

5. (2 pts) The following is a Python dictionary that we can use to make silly translations from English to French. The example is from the book Introduction to Computation and Programming using Python by John Guttag.

```
eng_to_fr = {"bread":"pain", "wine":"vin", "with":"avec", "I":"Je",
             "eat":"mange", "drink":"bois",  "John":"Jean",
"friends":"amis", "and":"et","of":"du","red":"rouge"}
```

   a. Write a Python expression that would
      1. Print all the English words found in our dictionary such that each word is printed on a separate line.

      2. Print all the French words found in our dictionary such that each word is printed on a separate line.

      3. Add the key,value pair ("white":"blanc") to the dictionary eng_to_fr .

   b. Write a Python function translate(s, d) where s stands for an English sentence represented as a list of strings and d stands for a dictionary, and returns a list of strings corresponding to the translation of the given sentence to French. Note: You will not be penalized on syntax errors since this is a written assignment. However, we recommend that you test your code by actually implementing it in Python.

c. Write an English sentence consisting of words that appear as a key in the dictionary eng_to_fr . Write a Python expression uses the function translate that would return the translation of this sentence and state what the returned value is in this case. You are not supposed to define a new function. Note that you can assume that sentences are represented using string lists.