CSE101 – Midterm Exam 2

19-Nov-2018 **Total points: 50** **Time allotted: 80 mins**

Name: _____ Student ID # _____

Instructions: Read the questions carefully before attempting to write the answer. Write the answers in the space provided below each question. Use of pencil is encouraged, so that you can erase and overwrite. Make sure that your handwriting is legible. Rough work sheet is provided at the end of answer sheet – which is to be used only for rough work, not for writing answers.

1. Assume a list is defined with this statement:
   >>> halogens = [’F’, ’Cl’, ’Br’, ’I’, ’At’]

   Explain how the list would be sorted by a call to isort. The easiest way to do this is to show the lines that would be displayed by the print statement that displays the sorted and unsorted regions. Here are the first two lines, to get you started: (3 pts)

   >>> isort(halogens)

   [’F’] [’Cl’, ’Br’, ’I’, ’At’]

   [’Cl’, ’F’] [’Br’, ’I’, ’At’]

   ['Br', 'Cl', 'F'] ['I', 'At']

   ['Br', 'Cl', 'F', 'I'] ['At']

   ['At', 'Br', 'Cl', 'F', 'I']

2. Below is a test list with 16 numbers. Show how this list would be sorted by a call to msort. The easiest way to do this is to show the groups before each round of merges.
   1 99 3 47 50 37 79 71 15 51 87 28 19 93 91 70 (6 pts)

   The initial groups are given as below:

   [1] [99] [3] [47] [50] [37] [79] [71] [15] [51] [87] [28] [19] [93] [91] [70]

   After calling merge_groups with size 1

   _[1 99] [3 47] [37 50] [71 79] [15 51] [28 87] [19 93] [70 91] _____

   After calling merge_groups with size 2

   _[1 3 47 99] [37 50 71 79] [15 28 51 87] [19 70 91 93]_____

   After calling merge_groups with size 4

   _[1 3 37 47 50 71 79 99] [15 19 28 51 70 87 91 93]_____

   After calling merge_groups with size 8

   _[1 3 15 19 28 37 47 50 51 70 71 79 87 91 93 99] _____

3. Write assignment statements that create dictionaries for the following sets of data:
   • The months of the year, using first three letters of month names as the keys and numbers from 1 to 12 as values.

   • The colors of the rainbow are VIOLET, INDIGO, BLUE, GREEN, YELLOW, ORANGE, and RED. Using the letters in the acronym VIBGYOR as keys and these colors as values, place them in the *colors* dictionary. (4 pts)

```
months = { 'jan': 1, 'feb': 2, 'mar': 3, 'apr': 4, 'may': 5, 'jun': 6, 'jul': 7, 'aug': 8, 'sep': 9, 'oct': 10, 'nov': 11, 'dec': 12}

colors = __{‘V’: ‘Violet’, ‘I’: ‘Indigo’, ‘B’: ‘Blue’, ‘G’: ‘Green’, ‘Y’: ‘Yellow’, ‘O’: ‘Orange’, ‘R’: ‘Red’}_____

_____
```

4. Complete the following function with recursion to convert the integer n to a string of binary representation. E.g. dec2bin(15) = 1111 (4 pts)

```
def dec2bin(n):
  if n < 0:
    _return 'Must be a positive integer'_____
  elif n == 0:
    __return '0'_____
  else:
    ___return dec2bin(n//2) + str(n%2) _____
```

5. Write iterative and recursive functions to reverse a list of elements. (6 pts)

```
def reverse(a):
        newlist = []
        for item in a:
            newlist = [item] + newlist
            print ("item = ", item, " newlist = ", newlist)
        return newlist


def recursiveReverse(a):

    #base
    if len(a) == 0:
        return []
    else:
        return [a[-1]] + recReverse(a[:-1])
```

6. Write a recursive function to sum elements in a list of numbers. (3 pts)

```
def sumList(a):
    # base
    if len(a) == 0:
        return 0
    return a[0] + sumList(a[1:])
```

7. Output analysis: For the following sub-questions, write the output of python code lines in the space provided.

A) Suppose a list a is defined with this statement:

>>> a = [11, 0, 6, 12, 7, 8, 3, 15, 4, 10]

How many comparisons will be made by the following searches using the linear search method? (4 pts)

a. isearch(a, 0) __-> 2_____

b. isearch(a, 3) __-> 7_____

c. isearch(a, 9) __-> 10_____

d. isearch(a, 10) __-> 10_____

B) Suppose a variable s has been defined with this assignment statement:

>>> s = "To be, or not to be, that is the Question:"

What will Python print for each of the following statements? (4 pts)

a. >>> print(s) _____->To be, or not to be, that is the Question:_____

b. >>> print(len(s)) _____->42_____

c. >>> print(s.split()) ____->['To', 'be,', 'or', 'not', 'to', 'be,', 'that', 'is', 'the', 'Question:']_____

d. >>> import string

>>> print(s.strip(string.punctuation)) ____->'To be, or not to be, that is the Question'_____

C) Suppose a dictionary object is defined with the following statement:

>>> d = {'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}

What will Python print as the value of the following expressions? (6 pts)

a. >>> len(d) _____-> 7_____

b. >>> d['X'] _____-> 10_____

c. >>> d['Z'] _____-> KeyError: 'Z'_____

d. >>> 'Q' in d _____-> False_____

e. >>> 5 in d _____-> False_____

f. >>> list(d.keys()) _____-> ['M', 'D', 'C', 'L', 'X', 'V', 'I']_____

8. Objective questions:

A) Suppose somelist = [1, 2, 0]. Which of the following will change somelist to [2, 1, 0]? (1 pt)

a. somelist.insert(somelist.pop(2), 0)
b. somelist.insert(somelist.pop(1), 0)
c. somelist.insert(0, somelist.pop(2))
d. somelist.insert(0, somelist.pop(1))

B) Which of the following algorithms uses the strategy divide and conquer? (1 pt)

a. Linear Search Algorithm
b. Luhn Algorithm
c. Binary Search Algorithm
d. Insertion Sort Algorithm

C) Which of the following statements is false? (1 pt)

a. The strategy for the linear search and insertion sort algorithms is the same: iterate over every location in the list and perform some operation.
b. For any list containing n items, binary search requires roughly *n* comparisons to find the target element.
c. A successful search in a binary search algorithm might return after the first comparison.
d. Merge Sort and Quicksort are two divide-and-conquer sorting algorithms.

D) Given a list a = [1,3,5,7,9,11] and target element to be 9; what are the mid values (corresponding array elements) in the binary search iterations? (1 pt)

a. 5 and 9
b. 7 and 10
c. 8 and 9
d. 8 and 10

E) Given a list a = [1, 2, 3, 5, 8, 13, 21, 34]. How many iterations required to find 2 using the binary search algorithm? (1 pt)

a. 1
b. 2
c. 3
d. 4

F) Match the searching/sorting algorithm and its best, worst and average complexity. (5 pts)