CS 115, Spring 2017 – Test 2

Name: ________________________________ Date: ___________________

Pledge: ____

Total: ___ / 75

Closed book: no textbook, no electronic devices, one sheet of paper with notes. Read each question carefully before answering, as there is **no partial credit for any response**! You may work out the solution to each question within the test itself, but **only your answers on this cover page will be graded**.

Question 1

a)	**False**	(2 points)
b)	**cab**	(3 points)

Question 2

a)	**00011100**	(5 points)
b)	**11110001**	(5 points)
c)	**00001101**	(5 points)

Question 3

a)	**[]**	(3 points)
b)	**current**	(3 points)
c)	**target – current** or **current – target**	(3 points)
d)	**L[1:]**	(3 points)
e)	**current**	(3 points)

Question 4

a)	**mymod2**	(2 points)
b)	**self.assertEqual(mymod2.closest_to([1, 7, 2, 90, 50], 52), 50)**	(4 points)
c)	**self.assertEqual(mymod2.closest_to([1, 7, 2, 90, 50], 7), 7)**	(4 points)

Question 5

a)	**1**	(2 points)
b)	**1**	(2 points)
c)	**1**	(2 points)
d)	**1**	(2 points)
e)	**1 or True**	(2 points)

Question 6

a)	**(True, [])**	(5 points)
b)	**(False, [])**	(5 points)
c)	**target – lst[0]**	(5 points)
d)	**[lst[0]] + use_it[1]**	(5 points)

Question 1 (5 points)
Consider the following code:

```
score = {}
score[(1, 2)] = 'a'
score[(3, 4)] = 'b'
score[(2, 1)] = 'c'
```

(a) What is printed on the screen after this statement has executed? (2 points)

```
print('a' in score and (1, 2) in score)
```

(b) What is printed on the screen after this statement has executed? (3 points)

```
print( score[(2, 1)] + score[(1, 2)] + score[(3, 4)] )
```

Question 2 (15 points)

(a) What is the binary representation of twenty eight (i.e., 28_{10})? Write it using exactly 8 bits.

(b) Using two's complement with exactly 8 bits, what is the binary representation of negative 15 (i.e., -15_{10})?

(c) Using your answers from (a) and (b), what is $28_{10} - 15_{10}$ in **binary**? Perform the operation using **addition in binary with 8 bits**.

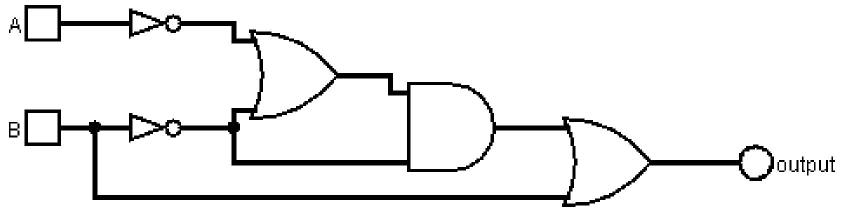
Question 3 (15 points)
Implement the following function, using recursion on L. That means you can access L only through the expressions L[0], L == [], and L[1:].

```
def closest_to(L, target):
    '''Assume L is a non-empty list and target is an integer.

    Returns the number in the list closest in value to the target value. If
    the target value is in the list, it should be returned. If there are
    multiple possibilities, return the first value in the list closest to the
    target.

    Example 1:
    closest_to([1, 7, 2, 90, 50], 52) => 50

    Example 2:
    closest_to([1, 7, 2, 90, 50], 7) => 7

    Example 3:
    closest_to([1, 7, 21, 90, 50], 4) => 1'''

    def closest_helper(L, current):

        if L == ___(a)___:
            return ___(b)___

        if abs(L[0] - target) < abs(________(c)________):
            return closest_helper(L[1:], L[0])

        return closest_helper(___(d)___, ___(e)___)

    return closest_helper(L[1:], L[0])
```

Question 4 (10 points)
Implement two PyUnit tests for the `closest_to` function you wrote in question 3. `test1` should cover example 1 from the docstring, and `test2` should cover example 2. Assume the `closest_to` function appears in module `mymod2`. Fill in the blanks in the PyUnit script. You must use assertEqual.

```
import unittest
import _____(a)_____ (2 points)

class Test(unittest.TestCase):

    def test1(self):
        _______________________________(b)___________________________________
        (4 points – no partial credit)


    def test2(self):
        _______________________________(c)___________________________________
        (4 points – no partial credit)

if __name__ == "__main__":
    unittest.main()
```

Question 5 (10 points)
Consider the following circuit:



Write out the truth table for this circuit. Your answers should output (0 or 1) for the given pair of inputs (A, B).

```
 A | B |          OUTPUTS
---+---+----------------------------
 0 | 0 |         ___(a)___
 0 | 1 |         ___(b)___
 1 | 0 |         ___(c)___
 1 | 1 |         ___(d)___
```

(e) Look at the truth table you just generated. Simplify the circuit to a SINGLE expression.

Question 6 (20 points)
Complete the implementation of this function for subset sum with the *"use it or lose it"* strategy.

```
subset_with_values(0, [27, 24, -1, 6, -2]) => (True, [])
subset_with_values(25, []) => (False, [])
subset_with_values(5000, [27, 24, -1, 6, -2]) => (False, [])
subset_with_values(25, [27, 24, -1, 6, -2]) => (True, [27, -2])
subset_with_values(50, [27, 24, -1, 6, -2]) => (True, [27, 24, -1])
```

```
def subset_with_values(target, lst):
    '''Determines whether or not it is possible to create the target sum
    using values in the list. Values in the list can be positive, negative, or
    zero. The function returns a tuple of exactly two items. The first is a
    Boolean that indicates True if the sum is possible and False if it's not.
    The second element in the tuple is a list of all the values that add up
    to make the target sum.'''
    if target == 0:
        return _____(a)______
    if lst == []:
        return _____(b)_______
    use_it = subset_with_values(__________(c)___________, lst[1:])
    if use_it[0]:
        return (use_it[0], __________(d)____________)
    return subset_with_values(target, lst[1:])
```

```
'''
CS 115 A, Spring 2017 - Test 2, Questions 7 and 8

Author: <your name here>
Pledge: <write pledge>
'''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' RULES: You can use Canvas to download this file and upload your solution.
' You can use Eclipse to edit and run your program. You should NOT look at
' other programs in Eclipse, you should NOT use any other programs, and you
' should NOT use any notes or books.
' According to the Honor Code, you should report any student who appears
' to be violating these rules.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Question 7 (10 points)
' Implement this function using recursion.
' Tribonacci numbers follow a pattern similar to Fibonacci numbers, except
' that the next number in the sequence is the sum of the previous three
' numbers.
' The sequence is as follows: 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, ...
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
def tribonacci(n):
    '''Returns the nth Tribonacci number. The 0th number is 0, the 1st number
    is 0, the 2nd number is 1, and any number beyond that is the sum of the
    previous three numbers.
    Examples:
    tribonacci(0) -> 0
    tribonacci(2) -> 1
    tribonacci(5) -> 4'''
    if n <= 1:
        return 0
    if n == 2:
        return 1
    return tribonacci(n - 1) + tribonacci(n - 2) + tribonacci(n - 3)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Question 8 (20 points)
' Implement this function using recursion AND memoization.
' Tribonacci numbers follow a pattern similar to Fibonacci numbers, except
' that the next number in the sequence is the sum of the previous three
' numbers.
' The sequence is as follows: 0, 0, 1, 1, 2, 4, 7, 13, 24, 44, ...
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
def trib_memo(n):
    '''Returns the nth Tribonacci number. The 0th number is 0, the 1st number
    is 0, the 2nd number is 1, and any number beyond that is the sum of the
    previous three numbers. Uses memoization to improve performance.'''
    def trib_helper(n, memo):
        if n in memo:
            return memo[n]

        if n <= 1:
            result = 0
        elif n == 2:
            result = 1
        else:
            result = trib_helper(n - 1, memo) + \
                     trib_helper(n - 2, memo) + \
                     trib_helper(n - 3, memo)
        memo[n] = result
        return result

    return trib_helper(n, {})
```