CS 111 first name: Xuan last name: Hu
Quiz 2 BU ID: U26611932

1. (5 points) Consider the following recursive function:

```
def mystery(x, y):
    if y > x:
        return y - 1
    print(x, y)
    return 3 + mystery(x // 2, y + 1)
```

a. What output will be produced by the call `mystery(16, 0)`?

```
16 0
8 1
4 2
```

b. What is the result (return value) produced by the call `mystery(16, 0)`?

11

2. (5 points) Write a recursive function `sum_odds(lst)` that takes a list of integers `lst` and returns the sum of the odd integers integers in `lst`.

For example, `sum_odds([1, 4, 8, 5])` should return 6, because 1 and 5 are the only odd values in `[1, 4, 8, 5]`, and their sum is 6.

Hint: You will need the % operator.

```
def sum-odds(lst):
 """ takes a list of integers lst
  and returns the sum of the odd int
    in lst
  """
  if lst == []:
     return 0
  else:
     rest = sum_odds(lst[1:])
     if lst[0] % 2 == 1:
          return rest + lst[0]
     else
        return rest
```

3. (5 points) Write a recursive function named `double_vowels(s)` that takes a string `s` and returns a string in which all of the vowels (a, e, i, o, u) in `s` are "doubled" (i.e., replaced by two occurrences of themselves), and all non-vowels are left alone. For example:

   `double_vowels('about')` should return the string `'aabboouut'`

   `double_vowels('time')` should return the string `'tiimee'`

   You may assume that the input contains only lowercase letters and no spaces.

```
def double_vowels(s):
  """ takes a string s and double
   the vowels in string and return
  """
  if s == '':
     return ''
  else:
     rest = double_vowels(s[1:])
     if s[0] in ['a','e','i','o','u']:
        return s[0]*2 + rest
     else:
        return s[0] + rest
```

4. (5 points) Binary numbers:

a. Convert this 8-bit binary number to decimal, showing your work: 10101100

$$10101100 = 2^2 + 2^3 + 2^5 + 2^7$$
$$= 4 + 8 + 32 + 128$$
$$= 172$$

b. Add these 4-bit binary numbers, without converting to decimal: 1010 + 0111
Show all of your work

```
   11
   1010
 + 0111
 ------
  10001
```