CS 111 first name: Xuan last name: Hu

Quiz 3 BU ID: U26611932

1. (6 points) Evaluate each of the following expressions:

a. `[c*2 for c in 'python']`

'PPyytthhoonn'

b. `[y for y in range(8) if y % 4 == 1]`

[1, 5]

c. `[[len(s), s[-1]] for s in ['we', 'love', 'python']]`

[[2,'e'], [4,'e'], [6,'n']]

d. `min([[s[0], len(s)] for s in ['python', 'is', 'amazing!']])`

['a', 8]

e. `[chr(ord(c) + 2) for c in 'abcde']`

['c', 'd', 'e', 'f', 'g']

f. `[x >> 2 for x in [170, 52, 28] if (x >> 2) % 2 == 0]`

[42]

2. (4 points) Assume that you've been given a helper function called `count_spaces(s)` that takes a string `s` and returns the number of spaces (`' '`) in `s`. For example, `count_spaces('oh hello there')` would return 2, because there are 2 spaces in that string.

Write a function `most_spaces(lst)` that takes a list of strings `lst` and returns the element of `lst` with the most number of spaces.

For example, `most_spaces(['a b c', 'd e', 'f g h i'])` should return `'f g h i'`, because it has 3 spaces, which is the largest number of spaces of any string in the list. Your function should use `count_spaces()` (which you do ***not*** need to write), along with a list comprehension.

3. (2 points) What is the output of the following code fragment?

```
num = 5
while num > 2:
    num = num - 1
    print(num, end=' ')
```

4. (4 points) Write a function `count_below(lst, threshold)` that takes a list of integers `lst` and an integer `threshold`, and **uses a loop** to compute and return the number of integers in `lst` that are below (i.e., less than) the specified `threshold`. For example, `count_below([1, 4, 8, 3, 10], 8)` should return 3, because there are 3 integers in that list (1, 4, and 3) that are below 8.

5. (4 points) Write a function `alternates(lst)` that returns `True` if the list of integers `lst` alternates between even and odd numbers (either [even, odd, even, odd, …] or [odd, even, odd, even, …]), and `False` otherwise. For example:

- `alternates([3, 2, 5])` and `alternates([4, 7, 2])` should both return `True.`
- `alternates([5, 8, 6, 7])` should return `False` because 8 and 6 are both even, and they are next to each other in the list.
- `alternates([3, 7, 8, 1])` should return `False` because 3 and 7 are both odd, and they are next to each other in the list.

You must use a loop. You may assume that the list has at least 2 elements.
Hint: You may find it helpful to compare `(lst[i] % 2)` to `(lst[i+1] % 2)`.

2.
```
def most_spaces(lst):
    """ takes a list of string lst and returns
        the element of lst with the most number
        of spaces.
    """
    ans = [[count_spaces(s), s] for s in lst]
    return max(ans)[1]
```

3. 4 3 2

4.
```
def count_below (lst, threshold):
    """ take a list of integers lst and threshold,
        return integer below threshold 's number.
    """
    count = 0
    for x in lst:
        if x < threshold:
            count += 1
    return count.
```

```
5. def alternates(lst):
       """ return True if lst alternates between even and odd
           numbers, and False otherwise.
       """
       for i in range(len(lst)-1):
           if lst[i]%2 == lst[i+1]%2:
               return False
       return True.
```