Assignment for Class Computer Science for Big Data B

Basic Python functions, Points: 13

1.) Given is a list of numbers:

```
numbers = [ 1, 4, 6, 67, 6, 8, 23, 8, 34, 49, 67, 6, 8, 23, 37, 67, 6, 34,
                19, 67, 6, 8 ]
```

a. Print the first four numbers (use slicing).

b. Print the last four numbers.

c. Write a Python function: *sum_all(_numbers: []) -> int* that computes the sum of all numbers.

d. Write a Python function: *sum_odds(…)* that computes the sum of all odd numbers.

e. Write a Python function: *sum_four(…)* that computes the sum of the first four numbers and the sum of the last four numbers. Use Python's tuple-notation to return both values.

f. Write a lambda function: (`sum_lambda = lambda` …) that computes the sum of all numbers.

g. Write a lambda function: *sum_lambda_odds(…)* that computes the sum of all odd numbers.

h. Given the list contains *n* numbers, how many steps are needed to compute the sum of all numbers?

You can reuse prior, own functions to build new functions, but not library functions. [6 Pts]

2.) Write a Python function: `find_first(_n: int, _numbers: []) -> int:`
that finds the first index of a number n in numbers (or -1 if element is not found).

Write a Python function: `find_last(_n: int, _numbers: []) -> int:`
that finds the last index of an element in numbers (or -1). [2 Pts]

3.) Write a Python function: `find_firsttwo(_n: int, _m: int, _numbers: []) -> int:`
that finds the first index of two given consecutive numbers in the list (or -1 if not found). [1 Pts]

4.) Write a Python function: `find_alltwo(_n: int, _m: int, _numbers: []) -> []:`
that finds all indices of the two given consecutive numbers appearing in the list (or empty set). [1 Pts]

5.) Write a Python function: `find_sub(_sub: [], _numbers: []) -> int:`
that finds the index of the first appearance of the sublist in the list (or -1). [1 Pts]

6.) Write a Python function: `find_allsubs(_sub: [], _numbers: []) -> []:`
that finds all occurrences of a given sublist in numbers (or empty set). [1 Pts]

7.) If numbers has *n* elements and the sublist has *m* elements, how many steps are needed to finds all occurrences of a sublist in the list? [1 Pts]

To test your functions, run

```
print("sum_all(numbers) ->         " + str( sum_all( numbers ) ) )
print("sum_odds(numbers) ->        " + str( sum_odds( numbers ) ) )
print("sum_four(numbers) ->        " + str( sum_four( numbers ) ) )
print("sum_lambda(numbers) ->      " + str( sum_lambda( numbers ) ) )
print("sum_lambda_odds(numbers) -> " + str( sum_lambda_odds( numbers ) ) )

print("find_first(6,numbers) ->           " + \
      str( find_first(6,numbers) ) )

print("find_last(6,numbers) ->            " + \
      str( find_last(6,numbers) ) )

print("find_firsttwo(6,8,numbers) ->      " + \
      str( find_firsttwo(6,8,numbers) ) )

print("find_alltwo(67,6,numbers) ->       " + \
      str( find_alltwo(67,6,numbers) ) )

print("find_sub([67,6,8],numbers) ->      " + \
      str( find_sub([67,6,8],numbers) ) )

print("find_allsubs([67,6,8],numbers) -> " + \
      str( find_allsubs([67,6,8],numbers) ) )
```

It should produce output:

```
numbers: [1, 4, 6, 67, 6, 8, 23, 8, 34, 49, 67, 6, 8, 23, 37, 67, 6, 34,
19, 67, 6, 8]

first four numbers: [1, 4, 6, 67]
last four numbers:  [19, 67, 6, 8]

sum_all(numbers) ->         554
sum_odds(numbers) ->        420
sum_four(numbers) ->        (78, 100)
sum_lambda(numbers) ->      554
sum_lambda_odds(numbers) -> 420

find_first(6,numbers) ->           2
find_last(6,numbers) ->            20
find_firsttwo(6,8,numbers) ->      4
find_alltwo(67,6,numbers) ->       [3, 10, 15, 19]
find_sub([67,6,8],numbers) ->      3
find_allsubs([67,6,8],numbers) -> [3, 10, 19]
```