

## Programming 2: Data Structures in MATLAB – Exercise 7

### Instructions

Submission date: 25/05/2020, 23:55

Submit 2 m-files:

**bubble\_sort.m, is\_complementary\_strand.m**

**Important!** It is your responsibility to verify your code is running properly. Clear all variables and run your code for each question in the command line to verify it runs properly.

You may use the files **init\_dll.m, bubble\_sort.m, dllnode.m, create\_dna\_sequence\_dll.m** and **get\_complementary\_nuc.m** provided for this exercise .

Please don't write `clear all;` command inside the files you submit.

**20 points will be deducted automatically from each file that prompts an error, no matter which.**

### Question 1

Change the **bubble\_sort.m** so that the result doubly linked list will contain:

- 1) All even numbers, sorted  
followed by:
- 2) All odd numbers, sorted

For example, suppose the input doubly linked list contained the following numbers:

7,4,10,11,3,33,200,333,6

The result doubly linked list should contain the following numbers in that order:

4,6,10,200,3,7,11,33,333

The original declaration should remain the same:

```
function bubble_sort (dll)
```

**Hint:** You can add an internal help function inside **bubble\_sort.m** file which decides if a pair of values should be swapped or not, and use it in the bubble\_sort function.

### Question 2

This question continues question 2 in tutorial 8. Write a function with the following declaration:

```
function result =  
is_complementary_strand(head_strand_1_dll,head_strand_2_dll)
```

The function accepts two **head** nodes for two different doubly linked lists. The function should return 1 if **head\_strand\_2\_dll** is the complementary strand of **head\_strand\_1\_dll** and 0 otherwise.

- You may assume that both dlls have the same size, and that both nodes are head nodes.
- You may use the function `get_complementary_nuc.m`. You may also use `create_dna_sequence_dll.m` for testing.

- You are not allowed to use the function `create_complementary_strand` seen in the tutorial.

-

You can test your answers using the following code:

### Question 1

```
array = [9 12 56 0 3254 45 97 3 2 15 1];
dll = init_dll(array);
bubble_sort(dll);

array = [7,4,10,11,3,33,200,333,6];
dll = init_dll(array);
bubble_sort(dll);

array = [7,4,10,11,3,33,200,333,10111];
dll = init_dll(array);
bubble_sort(dll);
```

**expected (FINAL) output:** (Notice that this is only the final output for each code section – there might be more lines printed, according to your implementation)

```
0 2 12 56 3254 1 3 9 15 45 97
4 6 10 200 3 7 11 33 333
4 10 200 3 7 11 33 333 10111
```

### Question 2

```
strand1 = create_dna_sequence_dll('AACG');
strand2 = create_dna_sequence_dll('CGTT');
strand3 = create_dna_sequence_dll('CGTA');
strand4 = create_dna_sequence_dll('TACG');

disp(is_complementary_strand(strand1, strand2));
disp(is_complementary_strand(strand2, strand1));
disp(is_complementary_strand(strand2, strand3));
disp(is_complementary_strand(strand1, strand3));
disp(is_complementary_strand(strand3, strand4));
disp(is_complementary_strand(strand2, strand4));
```

**expected output:**

```
1
```

```
1
```

0

0

1

0