

Assignment 2

Advanced Programming (INFO135)

Published at: 13:00, Friday, 11.02.2022

Deadline: 13:00, Friday, 18.02.2022

-
1. Suppose you have the following list of numbers to sort:

[1001, 1030, 1050, 1020, 300, 1080, 1100]

What will be the partially sorted list after 3 passes of Selection Sort?

-
2. Suppose you have the following list of numbers to sort:

[210, 15, 111, 90, 45, 120, 150, 200, 100, 140]

What will be the partially sorted list after 3 passes of Bubble Sort?

-
3. Write a function called `sort_and_rem_dup()` that receives a list of numbers and returns a **sorted** list where the **duplicates** in the numbers are removed.

Please note that:

- you can choose and implement **Selection** sort, **Insertion** sort or **Bubble** sort.
- you cannot use **Python Set** data structure to remove the duplicates.
- you cannot use **`sort()`** or **`sorted()`** built-in functions for Python list.

[Hint]: you can find an implementation of sorting algorithms in the slides of Lecture 3.

```
my_list = [5, 4, 3, 2, 1, 2, 3, 4, 5]
new_list = sort_and_rem_dup(my_list)
print(new_list)
```

[Output]:

[1, 2, 3, 4, 5]

4. Write a function `check_palindrome(word)` that receives a string variable called `word` as an input parameter, and builds a Stack and a Queue where their elements are the letters (characters) of that word. Then, the function should check and print if the word is a Palindrome or not.

[Hint 1]: Palindrome is a word whose characters are equal backward and forward.

[Hint 2]: you can find implementation of Stack and Queue in the slides of Lecture 2 & 3.

```
result = check_palindrome('hello')  
print(result)
```

```
result = check_palindrome('civic')  
print(result)
```

[Output]:

```
Not Palindrome  
Palindrome
```
