Indian Institute of Science Education and Research Mohali

Introduction to Computers (IDC101)
Academic Session 2018-19

Lab Session - 03 August 27-31, 2018

---

1. Write a function named swaplist that accepts the following inputs: (*a*). a list, say mylist, (*b*). two indices i and j in range(len(mylist)), and outputs a list which is same as mylist, except that elements at i^{th} and j^{th} position are now swapped. *Example:* swaplist(["i", "is", "e", "r"], 1, 3]) = ["i", "r", "e", "is"].

2. Use Exercise 1 to verify the following: "*All permutations of n symbols* $a_1, a_2, a_3, a_4, \cdots, a_n$ *can be obtained by repeated use of* swaplist". You may like to verify it for small values of n, say $n \in \{4, 5, 6\}$. Of course, the number of all permutations of n symbols is $n!$.

3. Use Exercise 1 to create a function named sortlist that accepts a list, say mylist, as input and outputs the list obtained after sorting mylist. *Example:* sortlist([32, 14, 67, 6, 79]) = [6, 14, 32, 67, 79]. Doesn't the fact that your swaplist based algorithm sorts any arbitrary list gives another perspective to Exercise 2?

4. Write a function isitsorted to determine if a given list is sorted.

5. Ask a user named *Careless Talkmore* to input names (along with surnames) of n students of your class. This user doesn't care if names are being input in proper case. For example, he is very likely to input name John Abraham as "joHn aBraham". Store the input in a list named ms18. Write a function named sortmyclass so that sortmyclass(ms18) returns the list of names sorted by ascending order of names. Also try writing a function sortmyclasssurname so that sortmyclasssurname(ms18) returns the list of names sorted in ascending order of surnames. Assume, for simplicity that each ms18[i] has excatly one blank space, and the surname starts following the blank space.

6. Write a function named primesupto that accepts an integer n as input and returns the list of all prime numbers less than or equal to n.

7. An integer is called *square free* if no perfect square except 1 divides it. The largest square free factor of an integer n is called the *square free part* of n. Write a function named sqrfree that accepts an integer n as input and outputs the square free part of n.
