

Birla Institute of Technology & Science, Pilani
2nd Semester 2014-15
Data Structures & Algorithms (CS/IS F211)
Lab-2 (Sorting)

Consider a parking lot which can accommodate fixed number of cars at a time. The parking lot is so huge, so that, to efficiently look-up whether the car is in the parking lot or not, we need to sort the list of cars based on their registration number. Each of these car is identified by a unique parking id, registration number [4 digit integer], and owner name.

Datastructures to be used –

- **Car** → This tuple (structure) maintains the details of the cars which includes its registration number and owner name.
- **ParkingLot** → This is a list of tuples <parking ID, Car>

Tasks to be performed –

- a) Create a header file *ParkingLot.h*
It will contain definitions for all the data structures used.
- b) Create a function *Enter_ParkingLot()*
This will insert a parking lot structure into an array of *ParkingLot*. The first step is to check for an overflow condition. Overflow means inserting a parking lot structure into an array which is full.
- c) Create a function *printCurrentlyParked()*
This function will print the currently parked cars in the non-decreasing order of registration numbers.
- d) Create a helper function *sortCars()*
This function will sort the list of cars using the below mentioned sorting algorithms.
- e) Complete the main function such that the number of lots available in the parking lot is taken as command-line input. After reading the input, it should print the cars available in sorted order.

Note –

Sort the list of cars based on registration number using –

1. Insertion Sort
2. Merge Sort

Take Home Assignment –

1. Implement the Parking lot as linked list, and perform sorting.
2. Implement the same using files for input and output.