Background

1. Slot no signifies the distance from the entrance.

Assumptions

1. customers are nice enough to always park in the slots allocated to them

Requirements

1. Support multi-story
2. Support commands
    a. Create_parking_lot
    b. park
    c. leave
    d. status
    e. registration_numbers_for_cars_with_colour $color
    f. slot_numbers_for_cars_with_colour $color
    g. slot_number_for_registration_number
3. Honor Capacity(no. cars) of the parking area
4. Ticket issuing process
    a. documenting the registration number (number plate) and the colour of the car
    b. allocate a parking slot that is nearest to the entry
    c. At the exit the customer returns the ticket which then marks the slot they were using as being available.
5. The system should be able to fetch/report
    a. registration number -> Slot number assigned
    b. color -> registration numbers of same color cars
    c. color -> Slot numbers of all slots where the same color cars are parked.
6. Shell-based interaction
    a. To run the program and launch the shell: $ my_program
    b. Assuming a parking lot with 6 slots, the following commands should be run in sequence by typing them in at a prompt and should produce output as described below the command:
        i. Input:
            1. create_parking_lot 6
        ii. Output:
            1. Created a parking lot with 6 slots
        iii. Input:
            1. park KA 01 HH 1234 White
        iv. Output:
            1. Allocated slot number: 1
        v. Input:
            1. park KA 01 HH 9999 White
        vi. Output:
            1. Allocated slot number: 2
        vii. Input:
            1. park KA 01 BB 0001 Black

viii. Output:
    1. Allocated slot number: 3

ix. Input:
    1. park KA 01 HH 7777 Red

x. Output:
    1. Allocated slot number: 4

xi. Input:
    1. park KA 01 HH 2701 Blue

xii. Output:
    1. Allocated slot number: 5

xiii. Input:
    1. park KA 01 HH 3141 Black

xiv. Output:
    1. Allocated slot number: 6

xv. Input:
    1. leave 4

xvi. Output:
    1. Slot number 4 is free

xvii. Input:
    1. Status

xviii. Output (we've used a table to make our lives easier, but tab-delimited output is fine):

Slot No.	Registration No	Colour
1	KA-01-HH-1234	White
2	KA-01-HH-9999	White
3	KA-01-BB-0001	Black
5	KA-01-HH-2701	Blue
6	KA-01-HH-3141	Black

xix. Input:
    1. park KA 01 P 333 White

xx. Output:
    1. Allocated slot number: 4

xxi. Input:
    1. park DL 12 AA 9999 White

xxii. Output:
    1. Sorry, parking lot is full

xxiii. Input:
    1. registration_numbers_for_cars_with_colour White

xxiv. Output:

        1. KA 01 HH 1234, KA 01 HH 9999, KA 01 P 333

    xxv. Input:
        1. slot_numbers_for_cars_with_colour White

    xxvi. Output:
        1. 1, 2, 4

    xxvii. Input:
        1. slot_number_for_registration_number KA 01 HH 3141

    xxviii. Output:
        1. 6

    xxix. Input:
        1. slot_number_for_registration_number MH 04 AY 1111

    xxx. Output:
        1. Not found

7. accept a filename as a parameter at the command prompt and read the commands from that file: my_program file_inputs.txt > output.txt
    a. To run the program: $ my_program file_inputs.txt > output.txt
    b. **Input** (in the file):
        i. create_parking_lot 6
        ii. park KA 01 HH 1234 White
        iii. park KA 01 HH 9999 White
        iv. park KA 01 BB 0001 Black
        v. park KA 01 HH 7777 Red
        vi. park KA 01 HH 2701 Blue
        vii. park KA 01 HH 3141 Black
        viii. leave 4
        ix. status
        x. park KA 01 P 333 White
        xi. park DL 12 AA 9999 White
        xii. registration_numbers_for_cars_with_colour White
        xiii. slot_numbers_for_cars_with_colour White
        xiv. slot_number_for_registration_number KA 01 HH 3141
        xv. slot_number_for_registration_number MH 04 AY 1111
    c. **Output** (to console, newline after every output):
        i. Created a parking lot with 6 slots
        ii. Allocated slot number: 1
        iii. Allocated slot number: 2
        iv. Allocated slot number: 3
        v. Allocated slot number: 4
        vi. Allocated slot number: 5

vii. Allocated slot number: 6
viii. Slot number 4 is free
ix. Slot No. Registration No Colour
1 KA 01 HH 1234 White
2 KA 01 HH 9999 White
3 KA 01 BB 0001 Black
5 KA 01 HH 2701 Blue
6 KA 01 HH 3141 Black
x. Allocated slot number: 4
xi. Sorry, parking lot is full
xii. KA 01 HH 1234, KA 01 HH 9999, KA 01 P 333
xiii. 1, 2, 4
xiv. 6
xv. Not found

8.

NFR

1. build+run on Linux
2. Without using any external libraries/gems except for a testing lib for TDD
3. Unit tests
4. coding conventions, directory structure and build approach - standard conventions
5. zip/tarball of your source code when you're done that includes Git metadata - we can look at your commit logs and understand how your solution evolved.