# Car Park Manager

Congratulations you have been appointed the manager of a small town Car Park!! This car park has a total of 10 available spaces in a prime area of town that’s within walking distance of the high street. As part of your new position you’ve decided to write a management system to offload the day-to-day activity so you can enjoy the finer things in life. After a quick brainstorm you’ve settled on the following functionality to make your life easier:

* Park – This should park a car in the first available space near the entrance, and provide a ticket number for that space. The ticket numbers will begin at 5000, should be non-reusable, and increment each time a new ticket is issued.
* Unpark – This should take a ticket number and unpark the car that’s in the space the ticket refers to.
* Compact – Eventually as cars are parked and unparked the available spaces will become fragmented, compacting the car park should move all the parked cars so that they take a contiguous set of spaces nearest the entrance.

In order to test your Car Park Manager you will need to take a string of comma separated commands from the stdin. Your application should process this command string and display the resultant free and taken status of each space in the car park at the end.

An example input is “pABC,pXYZ,pEFG,u5000,c” where:

* p means park the car with the given license plate.
* u means unpark the car in the space taken by that ticket
* c means compact the carpark

The expected output from this example would be “XYZ,EFG,,,,,,,,”