ROBOTICS – Python Development – Lab 4 - Homework

1. We need to implement an application that reads employee information from console using the following data fields:
    a. first_name: string, maximum 25 characters, non-empty;
    b. last_name: string, maximum 25 characters, non-empty;
    c. years_of_employment: integer, strictly positive, non-empty;
    d. is_employed: boolean, non-empty.

Please implement the following:

a. Read a list of employees from console performing the needed data validation, use dictionaries having as keys the field names. After each record that is read, ask the user if he wants to continue and read another record if so;
b. Save the read records into a CSV file.

2. Implement an export application that reads all records from the previously created CSV file and store them in a binary format. Please implement the following:

a. Store the records from the CVS file in a binary format into a file called export.dat.
b. Use the export.dat file to recreate the CSV data into a new file called export.csv.

Challenge problem

We need to simulate a solar powered drone robot which can store up to 200 energy points in its energy bank and having the following characteristics:

a. It flies linearly over 20 locations, it can only fly forward;
b. The drone moves forward after receiving the “move” command and it consumes 20 energy points to do so;
c. The drone can land using the “land” command, this consumes 10 energy points. While landed the drone cannot move;
d. Once landed, the drone can wait using the “wait” command. While waiting the drone recharges 50 energy points. The drone cannot store more energy than its energy bank allows;
e. The drone can take off by using the “takeoff” command, this consumes 10 energy points. This allows the drone to move again.

Please implement the following:

a. Start the drone at initial location landed and with energy bank full. Move the drone using manual commands to the final location where it should land. Avoid running run out of energy because this will crash the drone.
   The commands should be read from keyboard and the program should verify the consistency of commands (i.e. comand move will not be accepted if the drone is landed and the takeoff command will not be accepted if the drone is not landed).
b. At each command step, display the drone's information like in the following:

```
Energy left: 50 units
Landed: No

. . . . . . . . . . . . . . * . . . . .

Comand (move/land/wait/takeoff):_
```

   The drone is represented by the * character when flying and _ character when landed. The program stops once the drone lands at the final location or it crashes.

c. Generate the commands which will move the drone in an optimal way, this means using the minimum number of commands possible for landing at the final location.