COSC 1306
Spring 2021

Assignment 8: Lists and Functions

[1] **Objectives**: The primary purpose of this assignment is to practice list manipulation (again) and to learn some new features of functions (parameter passing, scope rules, local and global variables, etc.). You will write a program to use global variables and local variables. You will also return values to the calling function by using `return` or changing a list parameter.

[2] **Description**: We are building a simplified airline reservation system. There is only one flight that we are managing. The user (a reservation agent) makes a reservation for one seat by specifying a seat such as 3A. Unlike a real airline, we can only sell a seat once. Seats are divided into rows (starting at 1), and each row has a fixed number of seats (staring at A).

Data structure. You will be using a list of rows where a row is a list of letters. An example is given below.

```
seatTable = [
    ['X', '-', '-', '-'] # you can use a tuple for each row.
    ['-', '-', 'X', 'X']
] # for a plane with 2 rows and 4 seats per row.
```

I am using a nested list here. The nested list will give you some experience in using a 2-d array.

One of the issues that will cause problem for most students is the representation of a seat. On the user interface, people refer to the seats as 1A or 20F; the number starts at 1, and the letter starts at A. However, as most programming languages do, the internal representation of a seat is different. The index starts at 0. So the first row is stored in the 0-th element of the seat_table. This is nothing new and relatively easy to handle. The difference is one (minus one from external to internal). The letter part is slightly more complicated than the row number. The seats (A, B, C, D, E, F) are stored at index (0, 1, 2, 3, 4, 5) of the inner list. So you have to do the conversion. This type of conversion is very common in IT, human uses decimal system and computer use binary system, for example. I will show you how to do the conversion in class.

Here is what you need to do.

- Initialize the values to empty '−' and display the seat table.
- The program should have a loop that runs forever until all seats are taken.
- Allow the user to reserve an available seat (one at a time) by entering Row number and Column letter of their desired seat, such as '`1A`' or '`2B`'.
- If the seat exists and is available, change the seat status to RESERVED (from '−' to '`X`'). Otherwise, if the seat is already reserved or does not exist, display a proper error message.
- When all seats have been reserved, display a proper message to the user and then exit the program.

[3] **Requirements**:

I have been telling you NOT to use global variables. There is an exception. When you are sure the variable will not be changed anywhere in the program, it is (borderline) okay to use it. They are CONSTANTS. To make sure that you can spot a constant easily, it is the convention to use all uppercase variables (we see many of them in C/C++). Here are four constant that you are allowed to use. You are allowed to use these

to global variables only. You could have used '–'and 'X' in the code, but it is a good practice to give it a name.

```
AVAIL  = '-'
BOOKED = 'X'
```

Here is the list of functions that you have to implement. You will need to write some other functions to make the code better.

- `reset_table(seats)`: set the seat table to all AVAILable. "`seats`" is the list of lists. The function should return two values: the number of rows and the number of columns (`seats` per row). Do not return `seats`. This will force you to realize that lists such as `seats` can be changed inside a function.
- `get_res(num_row, num_col)`: get a reservation (for one seat) from the user and convert it to the "internal" representation (such as `1, A` to `0, 0`). Return the two indices if they are valid. If not, return `None, None`. You are required to use some local variables in this function.
- `reserve(seats, req_row, req_col)`: reserve a seat with the given row and column index. This function should NOT return anything, not even the seats. Any changes made to the seat table will be reflected in the main program.

`Seats` is the primary data structure that is used in almost all functions. Your main program should include a loop that runs until the flight is full. The rest of the requirements will be shown in the demonstration of the program during class.

Other comments:

- No global variables other than the two stated above.
- No nonlocal variables.
- The parameters used in the functions above should be enough for your use.

You have to make the code somewhat fault-tolerant (you should take care of the situation when the user typed in an incorrect seat request, like `0A` or `1X`). However, you may assume when asked for an integer number, and the user will enter an integer (of whatever value). Also, when asked for a letter, always accept both upper and lower cases. You can change the case.

[4] **Output**: See sample outputs below and the demo in class. A 2 x 2 airplane on the left and a 20 x 6 airplane on the right. Always test your code with a smaller case. In the 20 x 6 case, you will enter 240 values to cause the program to stop.

[5] **Extension**: It would be easier to enter seat selection as "8B" instead of two separate entries. You can try to do that. We will talk about strings later. Also, we are not checking errors such as the row number is not an integer. We will talk about exceptions that can handle such a situation soon.

>>> Go to the next page

```
Enter the number of rows: 2
Enter the number of seats per row: 2

Row A B
 1  - -
 2  - -

Enter a row number (1 to 2): 1
Enter a seat letter (A to B): a
Reservation request converted: (0,0)

Seat 1A booked.

Row A B
 1  X -
 2  - -

Enter a row number (1 to 2): 2
Enter a seat letter (A to B): a
Reservation request converted: (1,0)

Seat 2A booked.

Row A B
 1  X -
 2  X -

Enter a row number (1 to 2): 1
Enter a seat letter (A to B): a
Reservation request converted: (0,0)

Seat 1A is not available.

Row A B
 1  X -
 2  X -

Enter a row number (1 to 2): 1
Enter a seat letter (A to B): b
Reservation request converted: (0,1)

Seat 1B booked.

Row A B
 1  X X
 2  X -

Enter a row number (1 to 2): 2
Enter a seat letter (A to B): b
Reservation request converted: (1,1)

Seat 2B booked.

Row A B
 1  X X
 2  X X

The reservation if FULL for this flight.
```

```
Enter the number of rows: 20
Enter the number of seats per row: 6

Row A B C D E F
 1  - - - - - -
 2  - - - - - -
 3  - - - - - -
 4  - - - - - -
 5  - - - - - -
 6  - - - - - -
 7  - - - - - -
 8  - - - - - -
 9  - - - - - -
10  - - - - - -
11  - - - - - -
12  - - - - - -
13  - - - - - -
14  - - - - - -
15  - - - - - -
16  - - - - - -
17  - - - - - -
18  - - - - - -
19  - - - - - -
20  - - - - - -

Enter a row number (1 to 20): 10
Enter a seat letter (A to F): c
Reservation request converted: (9,2)

Seat 10C booked.

Row A B C D E F
 1  - - - - - -
 2  - - - - - -
 3  - - - - - -
 4  - - - - - -
 5  - - - - - -
 6  - - - - - -
 7  - - - - - -
 8  - - - - - -
 9  - - - - - -
10  - - X - - -
11  - - - - - -
12  - - - - - -
13  - - - - - -
14  - - - - - -
15  - - - - - -
16  - - - - - -
17  - - - - - -
18  - - - - - -
19  - - - - - -
20  - - - - - -

Enter a row number (1 to 20): 1
Enter a seat letter (A to F): D
Reservation request converted: (0,3)

Seat 1D booked.
```

	Row A B C D E F 1 - - - X - - 2 - - - - - - 3 - - - - - - 4 - - - - - - 5 - - - - - - 6 - - - - - - 7 - - - - - - 8 - - - - - - 9 - - - - - - 10 - - X - - - 11 - - - - - - 12 - - - - - - 13 - - - - - - 14 - - - - - - 15 - - - - - - 16 - - - - - - 17 - - - - - - 18 - - - - - - 19 - - - - - - 20 - - - - - - Enter a row number (1 to 20):

[5] **Deadline**: Midnight, Monday, April 12, 2021.