Download (right-click, save target as ...) this page as a jupyterlab notebook from: Lab4-HW

Laboratory 4: Input and Output

Medrano, Giovanni

R11521018

ENGR 1330 Laboratory 4 - Homework

In [1]:
```
# Preamble script block to identify host, user, and kernel
import sys
! hostname
! whoami
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
DESKTOP-6HAS1BN
desktop-6has1bn\medra
C:\Users\medra\anaconda3\python.exe
3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
```

Exercise 1 : Find the Treasure Part 1

Consider the structure below (a treasure map)

34	21	32	41	25
14	42	43	14	31
54	45	52	42	23
33	15	51	31	35
21	52	33	13	23

Background

Eventually we will write a program to explore the above matrix for a treasure, but first lets understand the map structure. The values in the array are clues.

Each cell in the matrix contains an integer between 11 and 55; for each value the ten's digit represents the row number and the unit's digit represents the column number of the cell containing the next clue.

For example starting in the upper left corner (at row = 1,column = 1), use the clues to guide your search of the array. So (1,1) contains 34. Next move to (3,4), which contains 43. Move to (4,2), which contains 15, and so on.

The treasure is a cell whose value is the same as its coordinates.

The program must eventually read in the treasure map data into a 5 by 5 array, and should output the cells it visits during its search, and a message indicating where you found the treasure.

The "Treasure Hunt Problem" is from the HackerRank.com avaiable at
https://www.hackerrank.com/contests/startatastartup/challenges/treasure-hunt

In [3]:
```
# script to generate the treasure map (do not modify)
treasuremap = [] # empty list to store row, column information
treasuremap.append([34,21,32,41,25]) # first row of the map
treasuremap.append([14,42,43,14,31]) # second row of the map
treasuremap.append([54,45,52,42,23])# third row of the map
treasuremap.append([33,15,51,31,35])# fourth row of the map
treasuremap.append([21,52,33,13,23])# fifth row of the map
```

Now for your problem Write a script to take input from user in terms of row number and column number. Return contents of treasure map at position indicated by the row and column to the user.

In [6]:
```
# script to access (1,3)  modify for interactive input
row = 5
column = 2
print(treasuremap[row-1][column-1]) # print a single element
```

```
52
```

Exercise 2 -- Following the Clues

Using your treasure map script and start at (1,1). Use the output clues as input and rerun the script. Stop when you have found the treasure. Here are a few to work with, remember to use the interactive input version that you created, you can add cells as needed, and cut and paste to generate the necessary sequence of cells

In [14]:
```
# script to access (1,1)  modify for interactive input
row, column = input("Enter the row and then the column:").split()
print(row, column)
row = int(row)
column = int(column)
# i don't really like doing it this way i couldve also used a map to do the same thing
print(treasuremap[row-1][column-1]) # print a single element
```

```
1 1
34
```

In [15]:
```
# script to access (3,4)  modify for interactive input
row, column = map(int, input("Enter the row and then the column").split())

#map(function, iterable, ...) from the python documentation the int function is applied
print(treasuremap[row-1][column-1]) # print a single element
```

```
42
```

In [17]:
```
# script to access (4,2)  modify for interactive input
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
15
```

In [18]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
25
```

In [19]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
31
```

In [20]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
54
```

In [21]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
13
```

In [22]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
32
```

In [23]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
45
```

In [24]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
35
```

In [25]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
23
```

In [26]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
43
```

In [27]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
51
```

In [28]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
21
```

In [29]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
14
```

In [30]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
41
```

In [31]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
33
```

In [32]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
52
```

In [33]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
52
```

In [34]:
```
row, column = map(int, input("Enter the row and then the column").split())
print(treasuremap[row-1][column-1]) # print a single element
```

```
52
```

WE HAVE OFFICIALLY ARRIVED TO THE TREASUREEEEE