Laboratory 4: Input and Output

LAST NAME, FIRST NAME

R00000000

ENGR 1330 Laboratory 4 - Homework

```
In [78]:  # Preamble script block to identify host, user, and kernel
          import sys
          ! hostname
          ! whoami
          print(sys.executable)
          print(sys.version)
          print(sys.version_info)
```

```
atomickitty
sensei
/opt/jupyterhub/bin/python3
3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0]
sys.version_info(major=3, minor=8, micro=10, 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 [90]:  # 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 [91]:   # script to access (1,3)  modify for interactive input
           row = 1
           column = 3
           print(treasuremap[row-1][column-1]) # print a single element
```

32

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 [102...   # script to access (1,1)  modify for interactive input
            row = 1
            column = 1
            print(treasuremap[row-1][column-1]) # print a single element
```

34

```
In [103...   # script to access (3,4)  modify for interactive input
            row = 3
            column = 4
            print(treasuremap[row-1][column-1]) # print a single element
```

42

```
In [104...   # script to access (4,2)  modify for interactive input
            row = 4
            column = 2
            print(treasuremap[row-1][column-1]) # print a single element
```

15

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js