# Assignment 2

**Due Date: Thursday April 6, 2023, at 11:59 PM**

## Problem Statement

1. At first write a Python program that reads the file "color_RGBs.txt" and creates a python dictionary `colorRBG` with

   - keys: the colors
   - values: the RGB values as a tuple

   for example, one item in the dictionary should be

```
Black : (0, 0, 0)
```

2. You will be given a one-dimensional List named **`colorList`** that contains several color names as strings then sort the `colorList` based on each color's RGB values using Lexicographic order and your dictionary `colorRGB` created in step 1.

   **As an example, given:**

```
colorList = ["Blue", "Pink", "Violet", "Olive", "Magenta", "Green"]
```

   The colors

   - "Blue" has RGB values (0, 0, 255)
   - "Pink" has RGB values (255,192,203)
   - "Violet" has RGB values (238,130,238)
   - "Olive" has RGB values (128,128,0)
   - "Magenta" has RGB values (255,0,255)
   - "Green" has RGB values (0, 128, 0)

   after sorting the `colorList` we should get:

```
colorList = ["Blue", "Green", "Olive", "Violet", "Magenta", "Pink"]
```

**3.** You will be given a two-dimensional List named **Board** which has some rows and columns. Each element in the Board, we call it a **cell**. Each cell will be assigned a **random color name** from the list named `colorList`.

For example,

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Magenta** | **Violet** | **Pink** | **Violet** | **Green** | **Blue** | **Green** | **Violet** |
| **Blue** | **Charcoal** | **Olive** | **Pink** | **Magenta** | **Pink** | **Magenta** | **Green** |
| **Green** | **Pink** | **Violet** | **Olive** | **Pink** | **Blue** | **Olive** | **Violet** |
| **Green** | **Magenta** | **Blue** | **Pink** | **Green** | **Pink** | **Magenta** | **Blue** |
| **Blue** | **Pink** | **Magenta** | **Green** | **Olive** | **Olive** | **Pink** | **Violet** |
| **Violet** | **Olive** | **Violet** | **Green** | **Pink** | **Magenta** | **Green** | **Pink** |
| **Pink** | **Blue** | **Green** | **Magenta** | **Blue** | **Violet** | **Blue** | **Green** |
| **Blue** | **Magenta** | **Olive** | **Violet** | **Violet** | **Charcoal** | **Pink** | **Olive** |

**4.** You will also be given a starting cell (row index and column index) in the Board. In the Board above, the cell whose cell is **shaded with yellow color** represents the starting cell.

**5.** You will be also given a **path** which is a one-dimensional list containing the movements: **forward**, **backward**, **upward,** or **downward**.

**For example, for a given path below:**

```
pathList = ["forward ", " forward ", "upward", "forward", " upward ", " upward
", "upward", "forward", "forward", "forward", "downward", "downward",
"downward", "backward"]
```

Our aim is to **traverse along the path** in the Board **starting from the starting cell** and **following the path** and **adding the cost of each cell we traverse** to compute the total cost of the path. The cost of each color is the index of the color in the **sorted** `colorList`.

In the example above the total cost will be

```
cost(Blue) + cost(Green) + cost(Magenta) + cost(Green) + cost(Pink) + cost(Olive)
+ cost(Green) + cost(Pink) + cost(Blue) + cost(Olive) + cost(Violet) + cost(Blue)
+ cost(Violet) + cost(Pink) + cost(Green)  = 0 + 1 + 4 + 1 + 5 + 2 + 1 + 5 + 0 +
2 + 3 + 0 + 3 + 5 + 1 = 33.
```

## Remark

- The total cost of the path is the sum of the costs of the cells along the path.
- The cost of a cell is the cost of the color name assigned to the cell.
- The cost of a color is the index of the color in the sorted list containing colors.
- In the example **`colorList`** above after sorting, Blue has cost 0, Green has cost 1, Olive has cost 2, ...., Pink has cost 5.

  **Note. In the actual code, the `colorList` is different from the example above, it is much bigger list**.

## Requirement

I have uploaded two files:

- **assignment2_starter_code.py**, and
- **color_RGBs.txt** (which contains the colors and their RGB values)

to create the `colorList`, the Board, set the starting cell, and the `pathList`. The program calls the functions below to do the required tasks. There are 6 functions in total the program:

1. `def createBoard(height, width, colorList):` ***This is already done for you.***
2. `def printBoard(Board, height, width):` ***This is already done for you.***
3. `def createPath(startRow, startColumn, height, width, pathLength):` ***This is already done for you.***
4. `def sortColorsRGB(colorList, colorRGB):` ***Implement this function.***
5. `def computeColorCost(colorList, colorName):` ***Implement this function.***
6. `def computePathTotalCost(pathList, startRow, startColumn, Board, colorList):` ***Implement this function.***