Ocean Lu
CS 4080.02
Professor Yang
10/29/19

Assignment #5

Task 1: https://colab.research.google.com/drive/1eHqLmLP3Ni0UaR-MKmJCFfAU6qsJCyY6

Program source code: Python

```python
# Initialize the input data in the program instead of entering from keyboard or reading from file.
import pandas as pd

df = pd.DataFrame([["Dan", 45],["Adam", 39],["Fiona", 42],["Kathy", 44],["Dan", 34],["Adam", 41],
                   ["Zehr", 43], ["Mona", 42], ["Kevin", 35], ["Elma", 48]], columns=list('AB'))
print("Original Data\n", df,"\n")

# (1) sort in ascending order of names, if same name ascending order of scores;
part1 = df.sort_values(by=['A','B'], ascending=[True,True])
print("Sort in ascending order of names, if same name ascending order of scores\n", part1, "\n")

# (2) sort in ascending order of names, if same name descending order of scores;
part2 = df.sort_values(by=['A','B'], ascending=[True,False])
print("Sort in ascending order of names, if same name descending order of scores\n", part2, "\n")

# (3) sort in descending order of names, if same name descending order of scores.
part3 = df.sort_values(by=['A','B'], ascending=[False,False])
print("Sort in descending order of names, if same name descending order of scores\n", part3, "\n")
```

Output runs:

```
Original Data
        A   B
0     Dan  45
1    Adam  39
2   Fiona  42
3   Kathy  44
4     Dan  34
5    Adam  41
6    Zehr  43
7    Mona  42
8   Kevin  35
9    Elma  48
```

```
Sort in ascending order of names, if same name ascending order of scores
        A   B
1    Adam  39
5    Adam  41
4     Dan  34
0     Dan  45
9    Elma  48
2   Fiona  42
3   Kathy  44
8   Kevin  35
7    Mona  42
6    Zehr  43
```

```
Sort in ascending order of names, if same name descending order of scores
        A   B
5    Adam  41
1    Adam  39
0     Dan  45
4     Dan  34
9    Elma  48
2   Fiona  42
3   Kathy  44
8   Kevin  35
7    Mona  42
6    Zehr  43
```

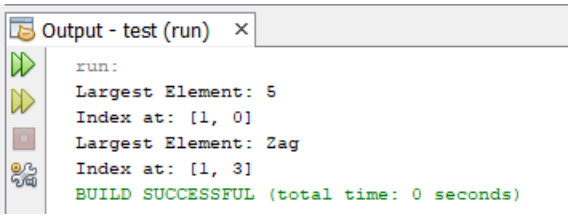
```
Sort in descending order of names, if same name descending order of scores
        A   B
6    Zehr  43
7    Mona  42
8   Kevin  35
3   Kathy  44
2   Fiona  42
9    Elma  48
0     Dan  45
4     Dan  34
5    Adam  41
1    Adam  39
```

Task 2:

Program Source Code: Java

```java
public class Test {

    static int findIntMax(int mat[][]) {
        int maxElement = mat[0][0];
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if (mat[i][j] > maxElement) {
                    maxElement = mat[i][j];
                }
            }
        }
        return maxElement;
    }

    static void findIntIndex(int mat[][], int maxElement) {
        boolean found = false;
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if (mat[i][j] == maxElement) {
                    System.out.println("Index at: [" + i +", " + j + "]");
                    found = true;
                    break;
                }
            }
            if (found) {
                break;
            }
        }
    }

    static String findStringMax(String mat[][]) {
        String maxElement = mat[0][0];
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if ((maxElement.compareTo(mat[i][j])) < 0) {
                    maxElement = mat[i][j];
                }
            }
        }
        return maxElement;
    }

    static void findStringMax(String mat[][], String maxElement) {
        boolean found = false;
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if (mat[i][j].equals(maxElement)) {
                    System.out.println("Index at: [" + i +", " + j + "]");
                    found = true;
                    break;
                }
            }
            if (found) {
                break;
            }
        }
    }

    public static void main(String[] args) {
        int integers[][] = {{1,2,4,4},{5,5,4,2},{3,1,1,5}};
        System.out.println("Largest Element: " + findIntMax(integers)) ;
        findIntIndex(integers, findIntMax(integers));

        String strings[][] = {{"Apple", "Banana", "Pork", "Beef"}, {"David",
            "Kelin", "Peter", "Zag", "Diana"}, {"Elin", "Adam", "Young",
                "Peter", "Zag"}};
        System.out.println("Largest Element: " + findStringMax(strings));
        findStringMax(strings, findStringMax(strings));
    }
```

Output:



Tests:

Test to find the max value in a 2D array:

```
16  public class Test {
17
18      static int findMax(int mat[][]) {
19          // Initializing max element as INT_MIN
20          int maxElement = Integer.MIN_VALUE;
21          // checking each element of matrix
22          // if it is greater than maxElement,
23          // update maxElement
            for (int i = 0; i < mat.length; i++) {
25              for (int j = 0; j < mat.length; j++) {
26                  if (mat[i][j] > maxElement) {
27                      maxElement = mat[i][j];
28                  }
29              }
30          }
31          // finally return maxElement
32          return maxElement;
33      }
34
35      public static void main(String[] args) {
36          int mat[][] = { { 1, 2, 3, 4 },
37                          { 25, 6, 7, 8 },
38                          { 9, 10, 11, 12 },
39                          { 13, 14, 15, 16 } };
40
41          System.out.println(findMax(mat)) ;
42      }
```

```
Output - test (run)
run:
25
BUILD SUCCESSFUL (total time: 0 seconds)
```

Test to find index (all), will be edited to return the first:

```java
    static void findIndex(int mat[][], int maxElement) {
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat.length; j++) {
                if (mat[i][j] == (maxElement)) {
                    System.out.println("Index at: [" + i +", " + j + "]");
                }
            }
        }
    }

    public static void main(String[] args) {
        int mat[][] = { { 1, 25, 3, 4 },
                        { 25, 6, 7, 8 },
                        { 9, 10, 11, 12 },
                        { 13, 14, 15, 16 } };

        System.out.println(findIntMax(mat)) ;
        findIndex(mat, findIntMax(mat));
    }
```

```
Output - test (run)
run:
25
Index at: [0, 1]
Index at: [1, 0]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Testing given integer data:

```java
    public static void main(String[] args) {
        int integers[][] = {{1,2,4,4},{5,5,4,2},{3,1,1,5}};
        System.out.println("Largest Element:" + findIntMax(integers)) ;
        findIntIndex(integers, findIntMax(integers));
    }
```

```
Output - test (run)
run:
Largest Number:5
Index at: [1, 0]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Testing max value of String in 2D array:

```java
    static String findStringMax(String mat[][]) {
        String maxElement = mat[0][0];
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if ((maxElement.compareTo(mat[i][j])) < 0) {
                    maxElement = mat[i][j];
                }
            }
        }
        return maxElement;
    }

    public static void main(String[] args) {
        int integers[][] = {{1,2,4,4},{5,5,4,2},{3,1,1,5}};
        System.out.println("Largest Element: " + findIntMax(integers)) ;
        findIntIndex(integers, findIntMax(integers));

        String strings[][] = {{"Apple", "Banana", "Pork", "Beef"}, {"David", "Kelin", "Peter", "Zag", "Diana"}, {"Elin", "Adam", "Young", "Peter", "Zag"}};
        System.out.println("Largest Element: " + findStringMax(strings));
    }
```

```
Output - test (run)
run:
Largest Element: 5
Index at: [1, 0]
Largest Element: Zag
BUILD SUCCESSFUL (total time: 0 seconds)
```

Testing to find the index of the String:

```java
    static void findStringMax(String mat[][], String maxElement) {
        boolean found = false;
        for (int i = 0; i < mat.length; i++) {
            for (int j = 0; j < mat[i].length; j++) {
                if (mat[i][j].equals(maxElement)) {
                    System.out.println("Index at: [" + i +", " + j + "]");
                    found = true;
                    break;
                }
            }
            if (found) {
                break;
            }
        }
    }

    public static void main(String[] args) {
        int integers[][] = {{1,2,4,4},{5,5,4,2},{3,1,1,5}};
        System.out.println("Largest Element: " + findIntMax(integers)) ;
        findIntIndex(integers, findIntMax(integers));

        String strings[][] = {{"Apple", "Banana", "Pork", "Beef"}, {"David", "Kelin", "Peter", "Zag", "Diana"}, {"Elin", "Adam", "Young", "Peter", "Zag"}};
        System.out.println("Largest Element: " + findStringMax(strings));
        findStringMax(strings, findStringMax(strings));
    }
```

```
Output - test (run)
run:
Largest Element: 5
Index at: [1, 0]
Largest Element: Zag
Index at: [1, 3]
BUILD SUCCESSFUL (total time: 0 seconds)
```