Worksheet: Arrays and Loops – Solutions

2D arrays are helpful in storing information about data collected when a location needs to be stored. In archeology, a dig site is typically divided into a grid of one foot squares that are marked off with string.

The following array, grid, represents the number of artifacts found in each square of a dig site:

```
0 4 1 8 6 1

4 7 7 0 7 1

0 8 8 5 3 1

4 0 3 4 4 7

6 1 7 5 6 0
```

1. Write the code to print out the total number of square feet at the site.

```
System.out.println("Total square feet: " + grid.length *
grid[0].length);
```

2. Write the code to find the total number of artifacts found at this dig site.

```
int sum =0;
for (int i =0; i < grid.length; i++) {
      for (int j =0; j < grid[0].length; j++) {
            sum += grid [i][j];
      }
}
```

3. Write a method to count how many locations on the grid have a certain number of artifacts. So if 7 is passed in, it would return 5.

```
public int artifactCount (int grid[][], int count) {
      int sum =0;
      for (int i =0; i < grid.length; i++) {
            for (int j =0; j < grid [0].length; j++) {
                  if ( grid [i] [j] == count)
                        sum ++;
            }
      }
      return sum;
}
```

4. Write the code for a class, Location, to store the row and column of an artifact in the grid. The maximum size of the grid should be 500x500. The class should include the following methods:

- Location() – Default constructor initializes to location (0,0).
- Location(int r, int c) – Constructor to create a location with row r and column c.
- int getRow() and int getCol() – Getter methods which return the row and column of the location.
- void setRow(int r) and void setCol(int c) – Setter methods which modify the row and column of the location. (These should also make sure that the new location is within the maximum grid size)
- boolean equals(Location o) – Comparison method which returns true if two locations have the same row and column.

```
public class Location {
      private int row;
      private int col;

      public Location () {
            row = 0;
            col = 0;
      }

      public Location (int r, int c) {
            this();
            setRow(r);
            setCol(c);
      }

      public int getRow() {
            return row;
      }

      public int getCol() {
            return col;
      }

      public boolean equals (Location o) {
            return (row == o.row) && (col == o.col);
      }

      public void setRow(int r) {
            if (r >= 0 && r < 500)
                  row = r;
      }

      public void setCol (int c) {
            if (c >= 0 && c < 500)
                  col = c;
      }

}//class Location
```

5. Write a method to return an ArrayList of all locations that contain a certain number of artifacts. So if 5 is passed in, the method would return a list of the locations (2,3) and (4,3). (Use the location class that you wrote in question 4.)

```
public static ArrayList returnLocations (int grid[][], int count) {
       ArrayList <Location> a = new ArrayList <Location> ();
      for (int i = 0; i < grid.length; i++) {
            for (int j = 0; j < grid [0].length; j++) {
                  if (grid[i][j] == count)
                        a.add (new Location (i, j));
            }
      }
      return a;
}
```

6. Re-write the method from question 3, this time using the method and class that you wrote for questions 4 and 5.

```
public int artifactCount (int grid[][], int count) {
      return returnLocations(grid, count).size();
}
```