Sahil A. Nawab – Array Exercises

AltSum.java

```
package Arrays;

public class AltSum {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int[] data = {1, 4, 9, 16, 9, 7, 4, 9, 11};
		int sum = 0;

		for (int i = 0; i < data.length; i++) {
			if (i % 2 == 0) {
				sum += data[i];
			}
			else {
				sum -= data[i];
			}
		}

		System.out.println("Sum: " + sum);
	} // end main method

} // end class AltSum
```

BulgarianSolitaire.java

```
package Arrays;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class BulgarianSolitaire {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner keyBoard = new Scanner(System.in);

		System.out.print("Please enter max cards (must be a triangular
number): ");
		final int MAX_CARDS = keyBoard.nextInt();

		ArrayList<ArrayList<Integer>> cardArray =  new
ArrayList<ArrayList<Integer>>();
		int column = 1, card = 0, left = MAX_CARDS;

		// generate random deck
		do {
			column = (int) (Math.random() * left + 1);
			System.out.println("Column: " + column);

			ArrayList<Integer> row = new ArrayList<Integer>();
```

```
                for (int i = 0; i < column; i++) {
                card++;
                row.add(card);
              }

                cardArray.add(row);
                left = MAX_CARDS - card;
            } while (card < MAX_CARDS);

            boolean finished = false;

            while (!finished) {
                ArrayList<Integer> row = new ArrayList<Integer>();

                // remove one card from each row
              for (int i = 0; i < cardArray.size(); i++){
                row.add(cardArray.get(i).get(0));
                cardArray.get(i).remove(0); // remove the first column

                if (cardArray.get(i).isEmpty()) {
                    cardArray.remove(cardArray.get(i));
                    i--;
                }
              }

              cardArray.add(row);

              ArrayList<Integer> size = new ArrayList<Integer>(); // new list
              for (int i = 0; i < cardArray.size(); i++){
                size.add(cardArray.get(i).size());
              }

              Collections.sort(size);

              for (int i = 1; i < size.size(); i++){
                if (size.get(i) == size.get(i - 1) + 1) {
                      finished = true;
                }
                else {
                    finished = false;
                    break;
                }
              }

              for (int i = 0; i < cardArray.size(); i++) {
                Collections.sort(cardArray.get(i));
              }
            }

            // display result
            System.out.println(cardArray);

      } // end main method

} // end class BulgarianSolitaire
```

LatinSquare.java

```
package Arrays;

import java.util.Scanner;

public class LatinSquare {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner keyBoard = new Scanner(System.in);

		System.out.print("Please enter size of matrix: ");
		final int N = keyBoard.nextInt();

		int[][] matrix = new int[N][N];

		int value = 1;
		for (int i = 0; i < N; i++) {
			for (int j = 0; j < N; j++) {
				matrix[i][j] = value;
				if (value < N)
					value++;
				else
					value = 1;
			}

			if (value < N)
				value++;
			else
				value = 1;
		}

		// display matrix
		for (int i = 0; i < N; i++) {
			for (int j = 0; j < N; j++) {
				System.out.print(matrix[i][j] + "  ");
			}

			System.out.println(); // blank line for spacing
		}
	} // end main method

} // end class LatinSquare
```

LinearSearch.java

```
package Arrays;

import java.util.Scanner;

public class LinearSearch {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner keyBoard = new Scanner(System.in);
```

```
		System.out.print("Please enter length of array: ");
		int userIn = keyBoard.nextInt();

		int[] array = new int[userIn];

		// populate array with random numbers
		for (int i = 0; i < userIn; i++) {
			array[i] = (int) (Math.random() * 50) + 1;
		}

		System.out.print("Please enter guess: ");
		int userGuess = keyBoard.nextInt();

		// print array for user
		if (userIn <= 40) {
			for(int i = 0; i < userIn; i++) {
				System.out.print(array[i] + " ");
			}
		}
		else {
			System.out.println("Array is too large to be displayed.");
			System.out.println("All calculations will still be
performed.");
		}

		System.out.println(); // blank line for spacing

		int counter = 0;

		for (int i = 0; i < userIn; i++) {
			if (userGuess == array[i]) {
				System.out.println(userGuess + " is at position " +
i);
				counter++;
			}
		}

		if (counter == 0) {
			System.out.println(userGuess + " is not present in the
array.");
		}
	} // end main method

} // end class LinearSearch
```

MagicSquares.java

```
package Arrays;

import java.util.Scanner;

public class MagicSquares {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner keyBoard = new Scanner(System.in);
```

```
            System.out.print("Please enter size of magic square (must be
odd): ");
            int n = keyBoard.nextInt();

            if (n % 2 == 0) {
                  System.out.println("Please enter an odd integer.");
            }
            else {
                  int[][] array = new int[n][n];

                  // place first value
                   int row = n - 1;
                 int col = n / 2;
                 array[row][col] = 1;

                  // start algorithm
                for (int i = 2; i <= n * n; i++) {
                  if (array[(row + 1) % n][(col + 1) % n] == 0) {
                        row = (row + 1) % n;
                        col = (col + 1) % n;
                    }
                    else {
                        row = (row - 1 + n) % n;
                        // don't change col
                    }

                  array[row][col] = i;
                }

                  // print finished magic square
                System.out.println(); // blank line for spacing

                  for (int i = 0; i < n; i++) {
                        for (int j = 0; j < n; j++) {
                              if (array[i][j] < 10) {
                                    System.out.print("0" + array[i][j] + "
");

                              }
                              else {
                                    System.out.print(array[i][j] + " ");
                              }
                        }
                        System.out.println(); // blank line for spacing
                  }
            }
      } // end main method

} // end class MagicSquares
```

Match.java

```
package Arrays;

public class Match {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
```

```
		String[] a = {"aa", "bb", "cc"};
		String[] b = {"aaa", "xx", "bb"};
		String[] c = {"aaa", "b", "bb"};
		String[] d = {"", "", "ccc"};

		System.out.println("Matches: " + stringMatch(a, b));
		System.out.println("Matches: " + stringMatch(a, c));
		System.out.println("Matches: " + stringMatch(a, d));
	} // end main method

	public static int stringMatch(String[] first, String[] second) {
		int counter = 0;

		for (int i = 0; i < first.length; i++) {
			if (first[i].length() == 0 || second[i].length() == 0) {
				// continue without executing next if statement
			}
			else if (first[i].charAt(0) == second[i].charAt(0)) {
				counter++;
			}
		}

		return counter;
	}

} // end class Match
```

PositionSearch.java

```
package Arrays;

import java.util.Scanner;

public class PositionSearch {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Scanner keyBoard = new Scanner(System.in);

		System.out.print("Please enter length of array: ");
		int userIn = keyBoard.nextInt();

		int[] array = new int[userIn];

		// populate array with random numbers
		for (int i = 0; i < userIn; i++) {
			array[i] = (int) (Math.random() * 50) + 1;
		}

		// print array for user
		if (userIn <= 40) {
			for(int i = 0; i < userIn; i++) {
				System.out.print(array[i] + " ");
			}
		}
		else {
			System.out.println("Array is too large to be displayed.");
```

```
                  System.out.println("All calculations will still be
performed.");
            }

            System.out.println(); // blank line for spacing

            int counter = 0;

            for (int i = 0; i < userIn; i++) {
                  if (array[i] > 30) {
                        System.out.println("The first number greater than 30
(" + array[i] + ") is at position " + i);
                        counter++; break;
                  }
            }

            if (counter == 0) {
                  System.out.println("No numbers greater than 30 are present
in the array.");
            }
      } // end main method

} // end class PositionSearch
```

Runs.java

```
package Arrays;

import java.util.Scanner;

public class Runs {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            Scanner keyBoard = new Scanner(System.in);

            System.out.print("Please enter length of array: ");
            int userIn = keyBoard.nextInt();

            int[] array = new int[userIn];

            // populate array with random numbers
            for (int i = 0; i < userIn; i++) {
                  array[i] = (int) (Math.random() * 6) + 1;
            }

            if (userIn <= 40) {
                  for(int i = 0; i < userIn; i++) {
                        System.out.print(array[i] + " ");
                  }
            }
            else {
                  System.out.println("Array is too large to be displayed.");
                  System.out.println("All calculations will still be
performed.");
            }
```

```
			System.out.println(); // blank line for spacing

			// handle first and last case
			if (array[0] == array[1]) {
				System.out.print("(" + array[0] + " ");
			}
			else {
				System.out.print(array[0] + " ");
			}

			for (int i = 1; i < (userIn - 1); i++) {
				if (array[i] == array[i + 1] && array[i] != array[i - 1]) {
					System.out.print("(" + array[i] + " ");
				}

				if (array[i] == array[i - 1] && array[i] != array[i + 1]) {
					System.out.print(array[i] + ") ");
				}
				else {
					System.out.print(array[i] + " ");
				}
			}

			if (array[userIn - 2] == array[userIn - 1]) {
				System.out.print(array[userIn - 1] + ")");
			}
			else {
				System.out.print(array[userIn - 1] + " ");
			}
		} // end main method

} // end class Runspackage Arrays;

import java.util.Scanner;

public class Runs {

		public static void main(String[] args) {
			// TODO Auto-generated method stub
			Scanner keyBoard = new Scanner(System.in);

			System.out.print("Please enter length of array: ");
			int userIn = keyBoard.nextInt();

			int[] array = new int[userIn];

			// populate array with random numbers
			for (int i = 0; i < userIn; i++) {
				array[i] = (int) (Math.random() * 6) + 1;
			}

			if (userIn <= 40) {
				for(int i = 0; i < userIn; i++) {
					System.out.print(array[i] + " ");
				}
			}
			else {
```

```
                  System.out.println("Array is too large to be displayed.");
                  System.out.println("All calculations will still be
performed.");
            }

            System.out.println(); // blank line for spacing

            // handle first and last case
            if (array[0] == array[1]) {
                  System.out.print("(" + array[0] + " ");
            }
            else {
                  System.out.print(array[0] + " ");
            }

            for (int i = 1; i < (userIn - 1); i++) {
                  if (array[i] == array[i + 1] && array[i] != array[i - 1]) {
                        System.out.print("(" + array[i] + " ");
                  }

                  if (array[i] == array[i - 1] && array[i] != array[i + 1]) {
                        System.out.print(array[i] + ") ");
                  }
                  else {
                        System.out.print(array[i] + " ");
                  }
            }

            if (array[userIn - 2] == array[userIn - 1]) {
                  System.out.print(array[userIn - 1] + ")");
            }
            else {
                  System.out.print(array[userIn - 1] + " ");
            }
      } // end main method

} // end class Runs
```

Sieve.java

```
package Arrays;

import java.text.DecimalFormat;
import java.util.Scanner;

public class Sieve {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            System.out.println("Sieve of Eratosthenes");
            Scanner keyBoard = new Scanner(System.in);

            System.out.print("Please enter upper bounds: ");
            final int MAX = keyBoard.nextInt();

            // initialize to primes
        boolean[] primes = new boolean[MAX + 1];
```

```
        for (int i = 2; i < primes.length; i++) {
            primes[i] = true;
        }

        displayPrimes(computePrimes(primes));
    } // end main method

    public static boolean[] computePrimes(boolean[] primes) {
        for (int i = 2; i * i < primes.length; i++) {
        // if i is prime, mark multiples of i as not prime
        if (primes[i] == true) {
            for (int j = i; i * j < primes.length; j++) {
                primes[i * j] = false;
            }
        }
    }
        return primes;
    }

    public static void displayPrimes(boolean[] primes) {
        System.out.println(); // blank line for spacing
        System.out.println("Primes from 1 to " + (primes.length - 1) +
":");

        DecimalFormat decimalFormat = new DecimalFormat("0000");
        byte counter = 0;

        for (int i = 0; i < primes.length; i++) {
            if (primes[i] == true)  {
                System.out.print(decimalFormat.format(i) + " ");
                counter++;
            }

            if (counter == 20) {
                System.out.println(); // blank line for spacing
                counter = 0;
            }
        }
    }

} // end class Sieve
```

Stars.java

```
package Arrays;

import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class Stars extends Applet {

    public void paint(Graphics g) {
        int randX, randY, randScale;
        Color randColor;

        for (int i = 0; i < 15; i++) {
```

```
                randX = (int) (Math.random() * 500) + 100;
                randY = (int) (Math.random() * 500) + 100;
                randScale = (int) (Math.random() * 8);
                randColor = genRandColor();

                drawStar(g, randColor, randX, randY, 25 * randScale, 25 *
randScale);
            }
      } // end paint method

      public void drawStar(Graphics g, Color c, int x, int y, int width, int
height) {
          g.setColor(c);
          for(int i = 0; i < 5; i++) {
              int x1 = (int)(circleX(5, i) * (double) width) + x;
              int y1 = (int)(circleY(5, i) * (double) height) + y;
              int x2 = (int)(circleX(5, (i + 2) % 5) * (double) width) + x;
              int y2 = (int)(circleY(5, (i + 2) % 5) * (double) height) + y;
              g.drawLine(x1,y1,x2,y2);
          }
      } // end method drawStar

      public double circleX(int sides, int angle) {
          double num = (double) angle / (double) sides;
          return Math.cos(TWO_PI * num - HALF_PI);
      } // end method circleX

      public double circleY(int sides, int angle) {
          double num = (double) angle / (double) sides;
          return Math.sin(TWO_PI * num - HALF_PI);
      } // end method circleY

      public Color genRandColor() {
            int randRed = (int) (Math.random() * 256),
                  randGreen = (int) (Math.random() * 256),
                  randBlue = (int) (Math.random() * 256);

            return new Color(randRed, randGreen, randBlue);
      } // end method genRandColor

      private final double HALF_PI = Math.PI / 2.0;
      private final double TWO_PI = Math.PI * 2.0;

} // end class Stars
```

Statistics.java

```
package Arrays;

import java.util.Arrays;
import java.util.Scanner;

public class Statistics {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            Scanner keyBoard = new Scanner(System.in);
```

```
            System.out.print("Please enter number of values (less than 20):
");
            int userIn = keyBoard.nextInt();

            int[] array = new int[userIn];

            // populate array with random numbers
            for (int i = 0; i < userIn; i++) {
                  array[i] = (int) (Math.random() * 20);
            }

            // print array for user
            if (userIn <= 40) {
                  for (int i = 0; i < userIn; i++) {
                        System.out.print(array[i] + " ");
                  }

                  System.out.println(); // blank line for spacing

                  Arrays.sort(array);

                  // print sorted array for user
                  for (int i = 0; i < userIn; i++) {
                        System.out.print(array[i] + " ");
                  }
            }
            else {
                  Arrays.sort(array);

                  System.out.println("Array is too large to be displayed.");
                  System.out.println("All calculations will still be
performed.");
            }

            System.out.println(); // blank line for spacing

            // calculate mean
            int sumValues = 0;
            double average = 0;

            for (int i = 0; i < userIn; i++) {
                  sumValues += array[i];
            }
            average = (double) sumValues / userIn;

            // calculate median
            double median = 0, med1, med2;

            if (userIn % 2 == 0) { // even length

                  med1 = array[userIn / 2];
                  med2 = array[userIn / 2 - 1];

                  median = (med1 + med2) / 2;
            }
            else { // odd length
```

```
                median = array[userIn / 2];
            }

            // calculate mode
            int mode = 0, modeCount = 0, currCount = 0;

            for (int i = 0; i < userIn; i++) {
                currCount = 0;

                for (int j = 0; j < userIn; j++) {
                    if (array[i] == array[j]) {
                        currCount++;
                    }
                }

                if (currCount > modeCount) {
                    modeCount = currCount;
                    mode = array[i];
                }
            }

            // print out final values
            System.out.println("Mean: " + average);
            System.out.println("Median: " + median);
            System.out.println("Mode: " + mode);

        } // end main method

} // end class Statistics
```