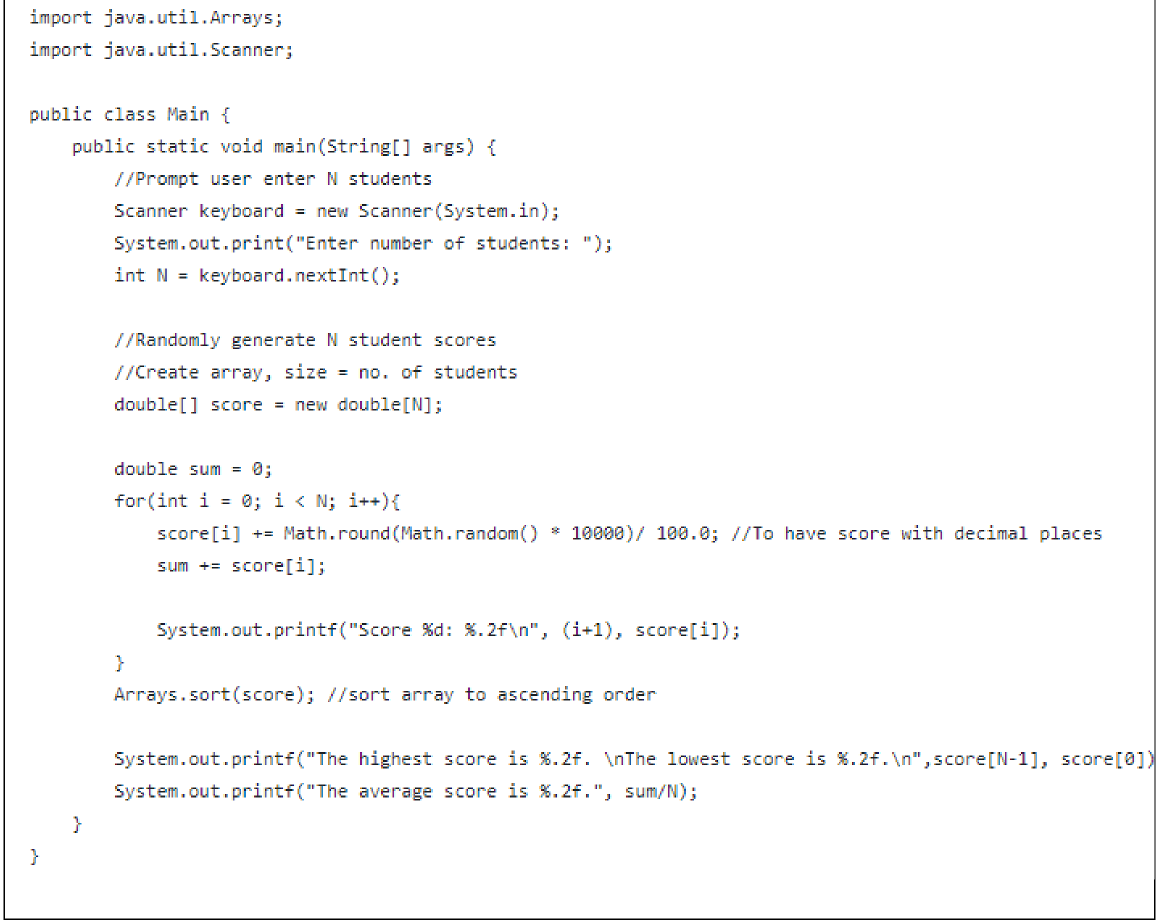
LAB 5

Arrays

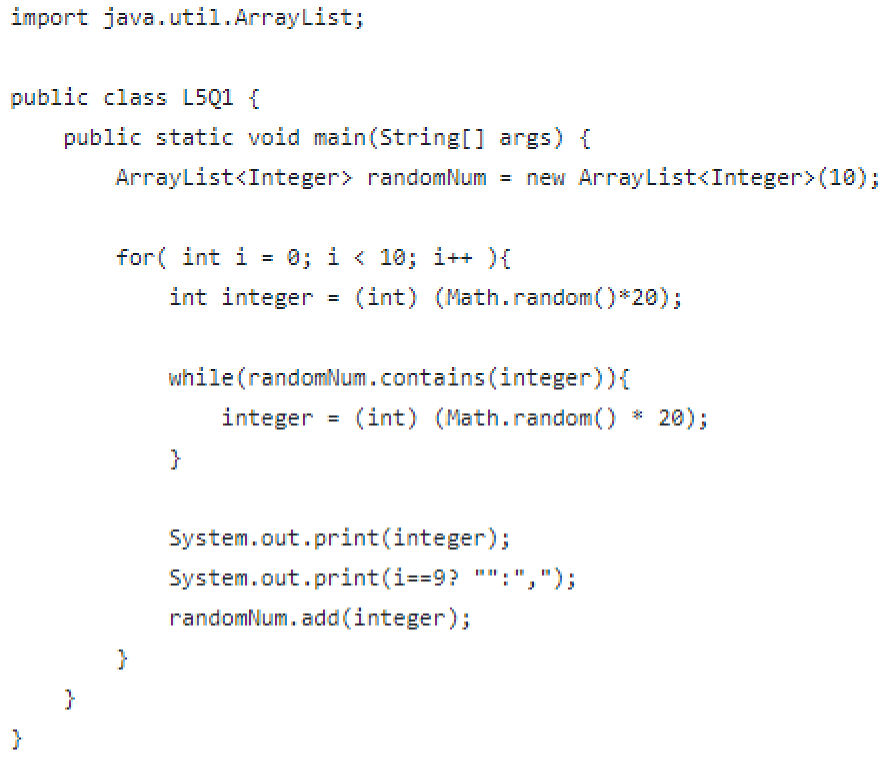
Name	:	Lee Pei Hui
Matric Number	:	22004781/1
Occurrence	:	7
Lecturer	:	Dr. Soraya

1. Write a program to randomly generate N student scores (0-100). The program will prompt the user to enter N students and store the score in an array. Then, the program will display a list of score, the highest score, the lowest score and the average score.



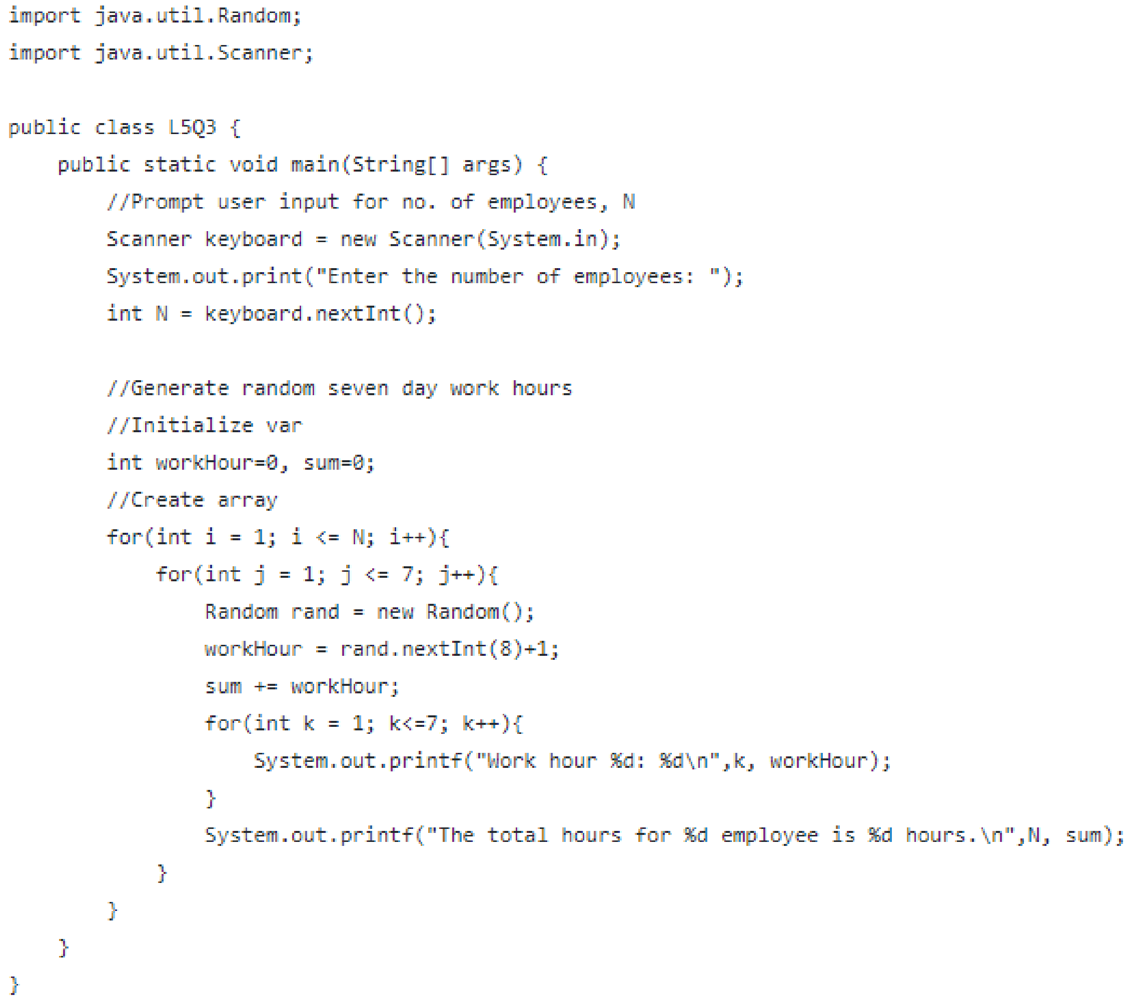
```
import java.util.Arrays;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        //Prompt user enter N students
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int N = keyboard.nextInt();

        //Randomly generate N student scores
        //Create array, size = no. of students
        double[] score = new double[N];

        double sum = 0;
        for(int i = 0; i < N; i++){
            score[i] += Math.round(Math.random() * 10000)/ 100.0; //To have score with decimal places
            sum += score[i];

            System.out.printf("Score %d: %.2f\n", (i+1), score[i]);
        }
        Arrays.sort(score); //sort array to ascending order

        System.out.printf("The highest score is %.2f. \nThe lowest score is %.2f.\n",score[N-1], score[0])
        System.out.printf("The average score is %.2f.", sum/N);
    }
}
```

2. Write a program that generates 10 non-duplicate random integers within the range from 0 to 20.



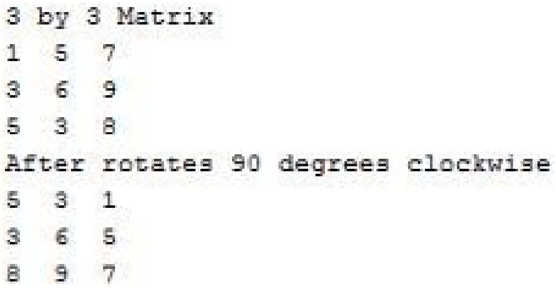
```
import java.util.ArrayList;

public class L5Q1 {
    public static void main(String[] args) {
        ArrayList<Integer> randomNum = new ArrayList<Integer>(10);

        for( int i = 0; i < 10; i++ ){
            int integer = (int) (Math.random()*20);

            while(randomNum.contains(integer)){
                integer = (int) (Math.random() * 20);
            }

            System.out.print(integer);
            System.out.print(i==9? "":",");
            randomNum.add(integer);
        }
    }
}
```

3. Write a program that randomly generate the seven day work hours (1-8 hours) for N employee. Then, display the work hours in seven days and the total hours for each employee.

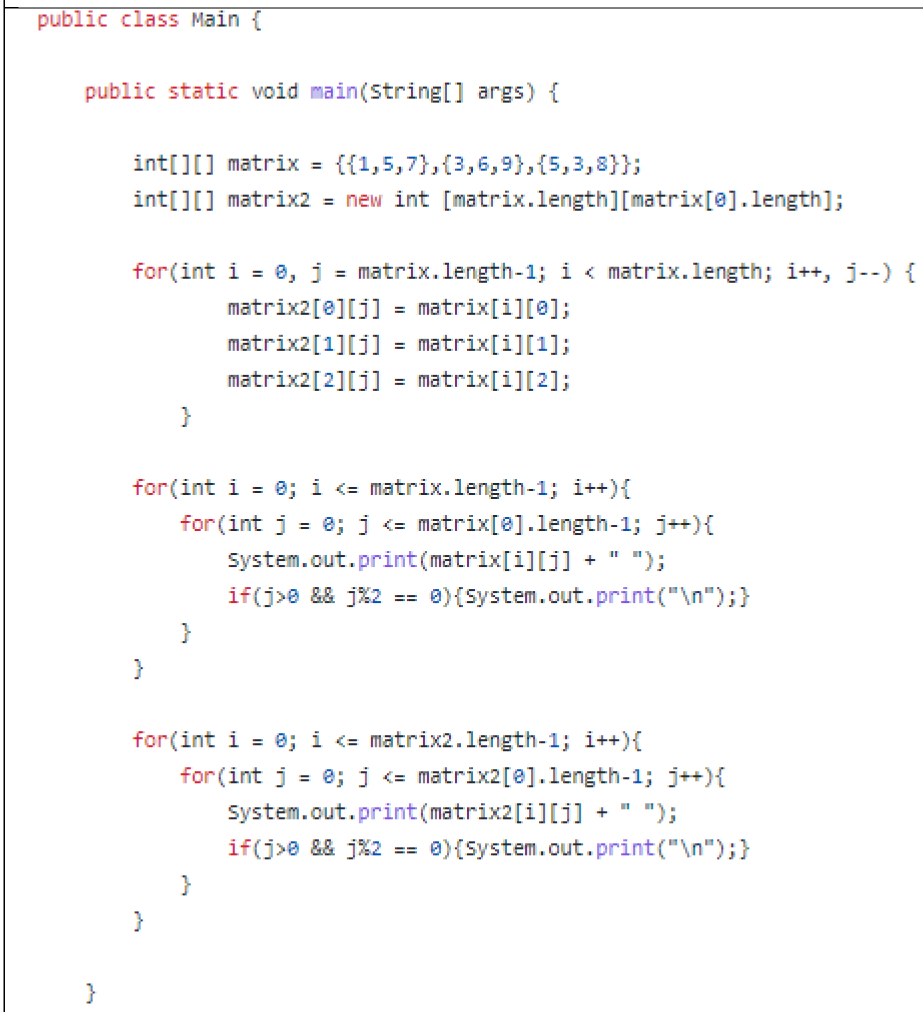


```
import java.util.Random;
import java.util.Scanner;

public class L5Q3 {
    public static void main(String[] args) {
        //Prompt user input for no. of employees, N
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter the number of employees: ");
        int N = keyboard.nextInt();

        //Generate random seven day work hours
        //Initialize var
        int workHour=0, sum=0;
        //Create array
        for(int i = 1; i <= N; i++){
            for(int j = 1; j <= 7; j++){
                Random rand = new Random();
                workHour = rand.nextInt(8)+1;
                sum += workHour;
                for(int k = 1; k<=7; k++){
                    System.out.printf("Work hour %d: %d\n",k, workHour);
                }
                System.out.printf("The total hours for %d employee is %d hours.\n",N, sum);
            }
        }
    }
}
```

4. Write a program that rotates 90 degrees clockwise a 3 by 3 matrix.

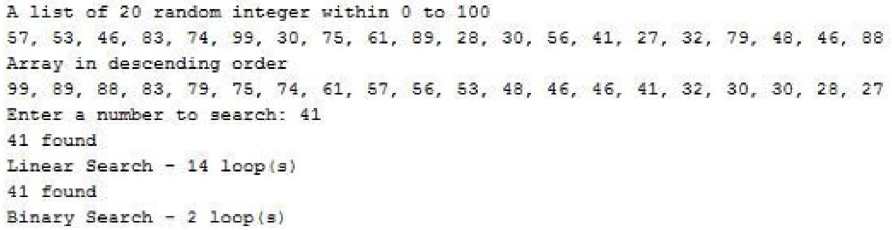


```
3 by 3 Matrix
1  5  7
3  6  9
5  3  8
After rotates 90 degrees clockwise
5  3  1
3  6  5
8  9  7
```



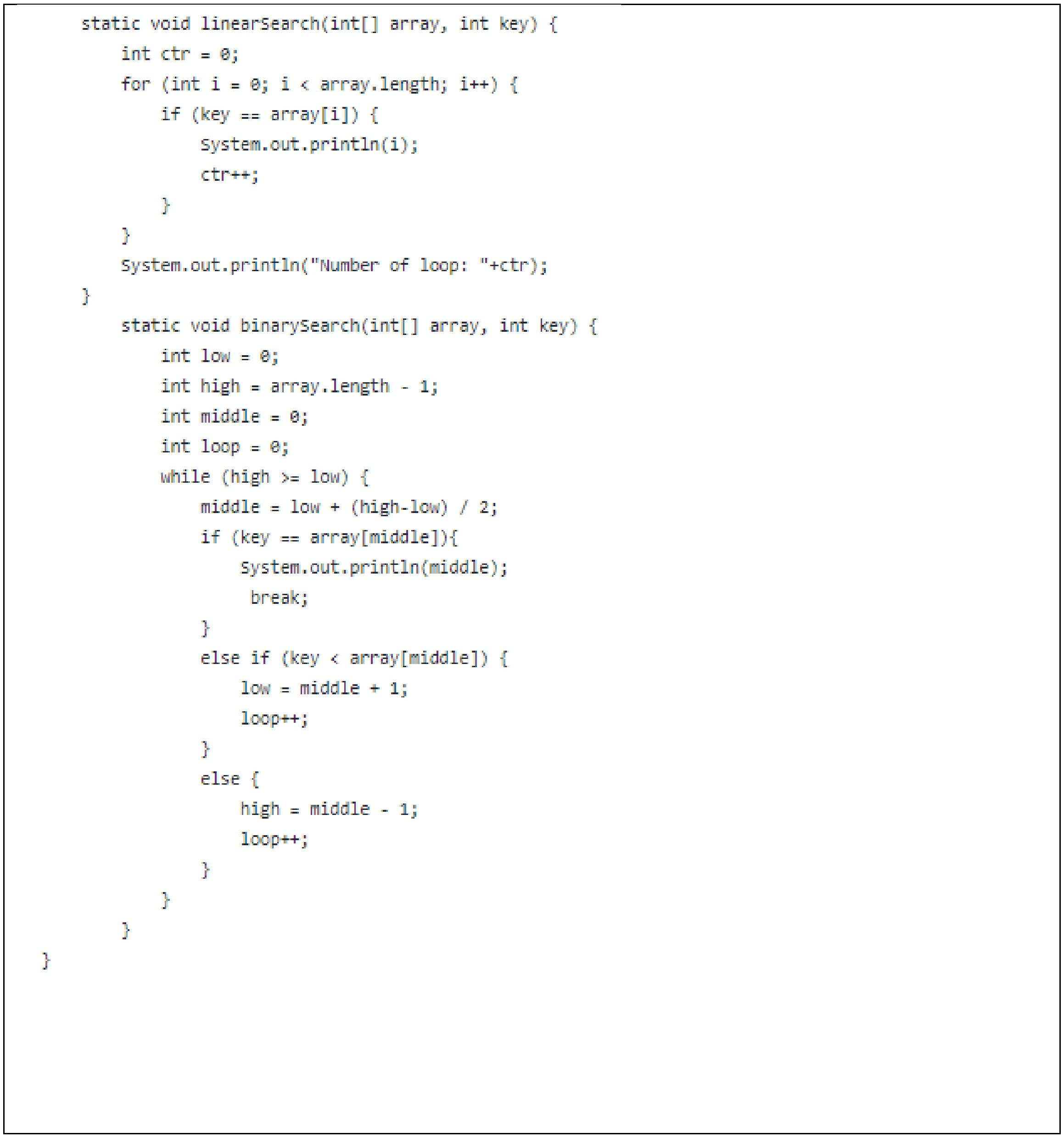
```java
public class Main {

    public static void main(String[] args) {

        int[][] matrix = {{1,5,7},{3,6,9},{5,3,8}};
        int[][] matrix2 = new int [matrix.length][matrix[0].length];

        for(int i = 0, j = matrix.length-1; i < matrix.length; i++, j--) {
                matrix2[0][j] = matrix[i][0];
                matrix2[1][j] = matrix[i][1];
                matrix2[2][j] = matrix[i][2];
            }

        for(int i = 0; i <= matrix.length-1; i++){
            for(int j = 0; j <= matrix[0].length-1; j++){
                System.out.print(matrix[i][j] + " ");
                if(j>0 && j%2 == 0){System.out.print("\n");}
            }
        }

        for(int i = 0; i <= matrix2.length-1; i++){
            for(int j = 0; j <= matrix2[0].length-1; j++){
                System.out.print(matrix2[i][j] + " ");
                if(j>0 && j%2 == 0){System.out.print("\n");}
            }
        }

    }
```

5. Write a program that generates 20 random integers within the range from 0 to 100. Sort the array in descending order. Then, accepts an integer input from the user. Then, search the array using this number. Compare the performance of linear search and binary search.



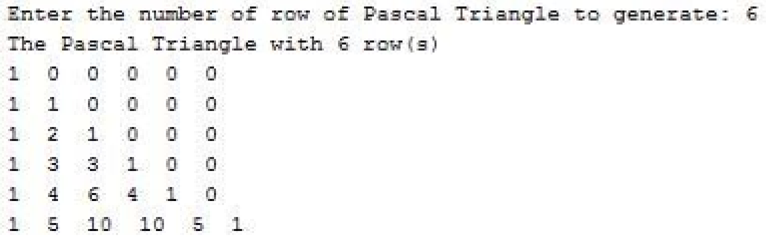
```
A list of 20 random integer within 0 to 100
57, 53, 46, 83, 74, 99, 30, 75, 61, 89, 28, 30, 56, 41, 27, 32, 79, 48, 46, 88
Array in descending order
99, 89, 88, 83, 79, 75, 74, 61, 57, 56, 53, 48, 46, 46, 41, 32, 30, 30, 28, 27
Enter a number to search: 41
41 found
Linear Search - 14 loop(s)
41 found
Binary Search - 2 loop(s)
```

```
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;

public class L5Q4 {

    public static void main(String[] args) {

        int[] arr = new int[20];

        System.out.println("A list of 20 random integer within 0 to 100");
        for (int i = 0; i < 20; i++) {
            arr[i] = (int) (Math.random() * 100);
            System.out.print(arr[i] + " ");
        }
        System.out.println();
        Arrays.sort(new int[][]{arr}, Collections.reverseOrder());

        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter a number to search: ");
        int search = keyboard.nextInt();

        linearSearch(arr, search);
        binarySearch(arr, search);
    }
```

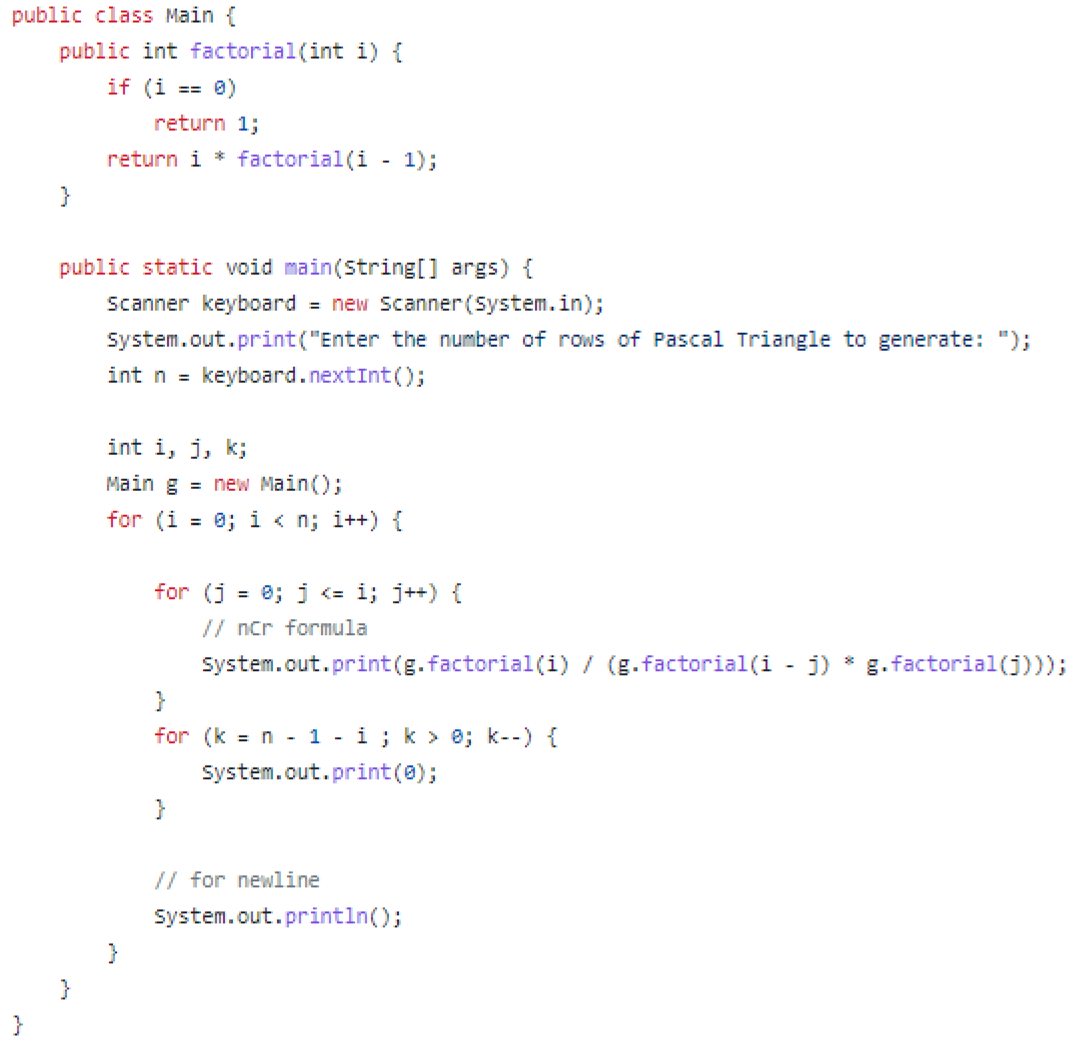


```
    static void linearSearch(int[] array, int key) {
        int ctr = 0;
        for (int i = 0; i < array.length; i++) {
            if (key == array[i]) {
                System.out.println(i);
                ctr++;
            }
        }
        System.out.println("Number of loop: "+ctr);
    }
        static void binarySearch(int[] array, int key) {
            int low = 0;
            int high = array.length - 1;
            int middle = 0;
            int loop = 0;
            while (high >= low) {
                middle = low + (high-low) / 2;
                if (key == array[middle]){
                    System.out.println(middle);
                     break;
                }
                else if (key < array[middle]) {
                    low = middle + 1;
                    loop++;
                }
                else {
                    high = middle - 1;
                    loop++;
                }
            }
        }
}
```

6. Write a program that used to create Pascal Triangle in the square matrix. The program will accept an integer from the users and use the integer to create the Pascal Triangle.



```
Enter the number of row of Pascal Triangle to generate: 6
The Pascal Triangle with 6 row(s)
1  0  0  0  0  0
1  1  0  0  0  0
1  2  1  0  0  0
1  3  3  1  0  0
1  4  6  4  1  0
1  5  10  10  5  1
```



```java
public class Main {
    public int factorial(int i) {
        if (i == 0)
            return 1;
        return i * factorial(i - 1);
    }

    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter the number of rows of Pascal Triangle to generate: ");
        int n = keyboard.nextInt();

        int i, j, k;
        Main g = new Main();
        for (i = 0; i < n; i++) {

            for (j = 0; j <= i; j++) {
                // nCr formula
                System.out.print(g.factorial(i) / (g.factorial(i - j) * g.factorial(j)));
            }
            for (k = n - 1 - i ; k > 0; k--) {
                System.out.print(0);
            }

            // for newline
            System.out.println();
        }
    }
}
```

Link:

https://github.com/PeiHui369/Fundamentals-Of-Programming/tree/main/Lab%205