9/7/24 **ASSIGNMENT-2**

1.Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2
5 3

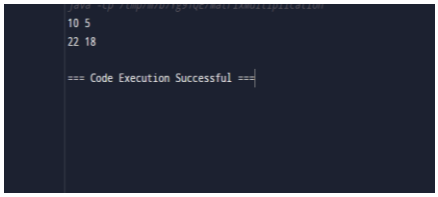
Mat2 = 2 3
4 1

Sample Output:

Mat Sum = 10 5
22 18

CODE:

```
public class MatrixMultiplication {

public static void main(String[] args) {

int[][] mat1 = {{1, 2}, {5, 3}};

int[][] mat2 = {{2, 3}, {4, 1}};

int[][] result = new int[2][2];

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

for (int k = 0; k < 2; k++) {

result[i][j] += mat1[i][k] * mat2[k][j];

}

}

}

for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) {

System.out.print(result[i][j] + " ");

}

System.out.println();

}

}
```



2.Write a program for matrix addition?

Sample Input:

Mat1 = 1 2
5 3

Mat2 = 2 3
4 1

Sample Output:

Mat Sum = 3 5
9 4

CODE:

```
public class MatrixAddition {
public static void main(String[] args) {
int[][] mat1 = {{1, 2}, {5, 3}};
int[][] mat2 = {{2, 3}, {4, 1}};
int rows = mat1.length;
int cols = mat1[0].length;
int[][] sum = new int[rows][cols];
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
sum[i][j] = mat1[i][j] + mat2[i][j];
}
}
System.out.println("Mat Sum = ");
for (int i = 0; i < rows; i++) {
for (int j = 0; j < cols; j++) {
System.out.print(sum[i][j] + " ");
}
System.out.println();
}
}
}
```

OUTPUT:

3.Write a program for Merge two sorted arrays using Array list

Input: arr1[] = { 1, 3, 4, 5}, arr2[] = {2, 4, 6, 8}

Output: arr3[] = {1, 2, 3, 4, 4, 5, 6, 8}

CODE:

```
import java.util.ArrayList;
import java.util.Collections;
```

```
public class MergeSortedArrays {
public static void main(String[] args) {
int[] arr1 = {1, 3, 4, 5};
int[] arr2 = {2, 4, 6, 8};
ArrayList<Integer> mergedList = new ArrayList<>();
for (int num : arr1) {
mergedList.add(num);
}
for (int num : arr2) {
mergedList.add(num);
}
Collections.sort(mergedList);
System.out.println("Merged Sorted Array: " + mergedList);
    }
    }
```

4.Find the Mean, Median, Mode of the array of numbers?

Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output:

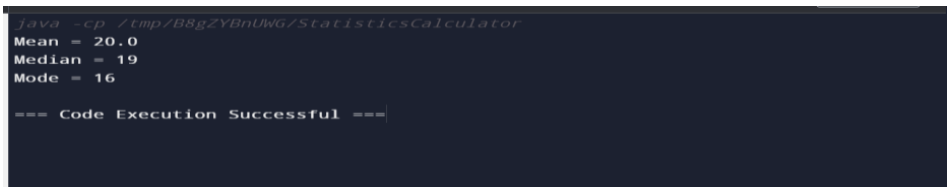
Mean = 20

Median = 19

Mode = 16

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;

public class StatisticsCalculator {

public static void main(String[] args) {
int[] numbers = {16, 18, 27, 16, 23, 21, 19};

double mean = calculateMean(numbers);
int median = calculateMedian(numbers);
int mode = calculateMode(numbers);

System.out.println("Mean = " + mean);
System.out.println("Median = " + median);
System.out.println("Mode = " + mode);
}

public static double calculateMean(int[] numbers) {
int sum = 0;
for (int num : numbers) {
sum += num;
```

```
}
return (double) sum / numbers.length;
}

public static int calculateMedian(int[] numbers) {
Arrays.sort(numbers);
int middle = numbers.length / 2;
if (numbers.length % 2 == 1) {
return numbers[middle];
} else {
return (numbers[middle - 1] + numbers[middle]) / 2;
}
}

public static int calculateMode(int[] numbers) {
Map<Integer, Integer> frequencyMap = new HashMap<>();
for (int num : numbers) {
frequencyMap.put(num, frequencyMap.getOrDefault(num, 0) + 1);
}

int mode = 0;
int maxFrequency = 0;
for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {
if (entry.getValue() > maxFrequency) {
mode = entry.getKey();
maxFrequency = entry.getValue();
}
}
return mode;
}
}
```



5.Write a program to find the number of composite numbers in an array of elements

Sample Input;:

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Sample Output:

Number of Composite Numbers = 5

Test cases:

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100, 100}

```
public class CompositeNumbersCounter;
public static void main(String[] args) {
int[] elements = {16, 18, 27, 16, 23, 21, 19};
int count = 0;
for (int num : elements) {
if (isComposite(num)) {
count++;
}
}
System.out.println("Number of Composite Numbers = " + count);
}
public static boolean isComposite(int num) {
if (num < 2) {
return false;
}
for (int i = 2; i <= num / 2; i++) {
if (num % i == 0) {
return true;
}
}
return false;
}
}
```

6.Write a program to print Right Triangle Star Pattern

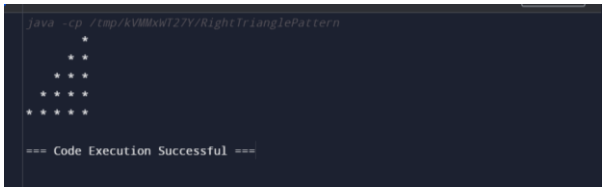
Sample Input:: n = 5

Output:

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
public class RightTrianglePattern {
public static void main(String[] args) {
int n = 5;
for (int i = 0; i < n; i++) {
for (int j = 0; j < n - i - 1; j++) {
System.out.print("  ");
}
for (int k = 0; k <= i; k++) {
System.out.print("* ");
```

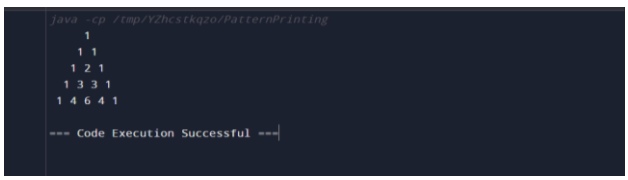
```
}
System.out.println();
}
}
}
```



7.Write a program to print the below pattern?

```
                1
            1       1
        1       2       1
    1       3       3       1
1       4       6       4       1
```

```
public class PatternPrinting {
public static void main(String[] args) {
int rows = 5;
for (int i = 0; i < rows; i++) {
int number = 1;
for (int j = rows; j > i; j--) {
System.out.print(" ");
}
for (int k = 0; k <= i; k++) {
System.out.print(number + " ");
number = number * (i - k) / (k + 1);
}
System.out.println();
}
}
}
```



8.Write a program to print rectangle symbol pattern.Get the symbol as input from user

```
import java.util.Scanner;
public class RectanglePattern {
public static void main(String[] args) {
```

```
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the symbol you want to use for the rectangle pattern: ");
char symbol = scanner.next().charAt(0);
System.out.println("Rectangle Pattern:");
for (int i = 1; i <= 5; i++) {
for (int j = 1; j <= 10; j++) {
System.out.print(symbol + " ");
}
System.out.println();
}
}
}
```

9.Write a program to print the following pattern

Sample Input:

Enter the number to be printed: 1
Max Number of time printed: 3
1
11
111
11
1

```
import java.util.Scanner;
public class PatternPrinter {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number to be printed: ");
int num = scanner.nextInt();
System.out.print("Max Number of times printed: ");
int max = scanner.nextInt();
for (int i = 1; i <= max; i++) {
for (int j = 1; j <= i; j++) {
System.out.print(num);
}
System.out.println();
}
for (int i = max - 1; i >= 1; i--) {
for (int j = 1; j <= i; j++) {
System.out.print(num);
}
System.out.println()  }
}
}
```

20.Write a program to print the Inverted Full Pyramid pattern?

```
public class InvertedFullPyramid {
   public static void main(String[] args) {
      int rows = 5;

      for (int i = rows; i >= 1; --i) {
         for (int space = 0; space < rows - i; ++space) {
            System.out.print("  ");
         }

         for (int j = i; j <= 2 * i - 1; ++j) {
            System.out.print("* ");
         }

         for (int j = 0; j < i - 1; ++j) {
            System.out.print("* ");
         }

         System.out.println();
      }
   }
}
```