11/7/24 **ASSIGNMENT-4**

1.Find the LCM and GCD of n numbers?

Sample Input:

N value = 2

Number 1 = 16

Number 2 = 20

Sample Output:

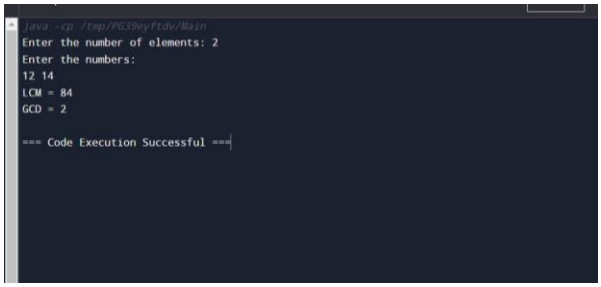
LCM = 80

GCD = 4

Test cases:

1. N = 3, {12, 25, 30}
2. N = 2, {52, 25, 63}
3. N = 3, {17, 19, 11}
4. N = -2, {52, 60}
5. N = 2, {30, 45}

```
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of elements: ");
int n = scanner.nextInt();
int[] numbers = new int[n];
System.out.println("Enter the numbers:");
for (int i = 0; i < n; i++) {
numbers[i] = scanner.nextInt();
}
int lcm = numbers[0];
int gcd = numbers[0];
for (int i = 1; i < n; i++) {
lcm = findLCM(lcm, numbers[i]);
gcd = findGCD(gcd, numbers[i]);
}
System.out.println("LCM = " + lcm);
System.out.println("GCD = " + gcd);
}
public static int findLCM(int a, int b) {
return a * b / findGCD(a, b);
}
public static int findGCD(int a, int b) {
if (b == 0) {
return a;
}
return findGCD(b, a % b);
}
}
```



2.Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. He is being offered 12 percent rate of interest; for all other customers, the ROI is 10 percent.

Sample Input:

Enter the principal amount: 200000
Enter the no of years: 3
Is customer senior citizen (y/n): n

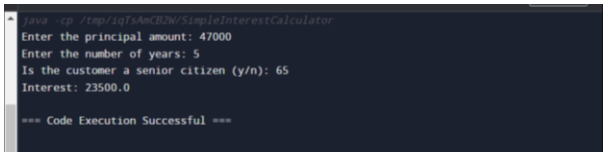
Sample Output:

Interest: 60000

Test Cases:

1. Principal: 2000 , Years: 0
2. Principal: 20000 , Years: -2
3. Principal: -2000 , Years: 2
4. Principal: 2 , Years: 2000
5. Principal: 0 , Years: 5

```
public class SimpleInterestCalculator {
public static void main(String[] args) {
double principal = 200000;
int years = 3;
char isSeniorCitizen = 'n';
double rateOfInterest = (isSeniorCitizen == 'y') ? 0.12 : 0.10;
double simpleInterest = (principal * rateOfInterest * years);
System.out.println("Simple Interest: " + simpleInterest);
}
}
```



3.Write a program to print the Fibonacci series.

Sample Input:

Enter the n value: 6

Sample Output:

0 1 1 2 3 5

```
public class FibonacciSeries {
public static void main(String[] args) {
int n = 6, firstTerm = 0, secondTerm = 1;
```

```
System.out.print("Fibonacci Series up to " + n + " terms: ");
for (int i = 1; i <= n; ++i) {
System.out.print(firstTerm + " ");
int nextTerm = firstTerm + secondTerm;
firstTerm = secondTerm;
secondTerm = nextTerm;
}
}
}
```

4.Java Program to Find Even Sum of Fibonacci Series Till number N?

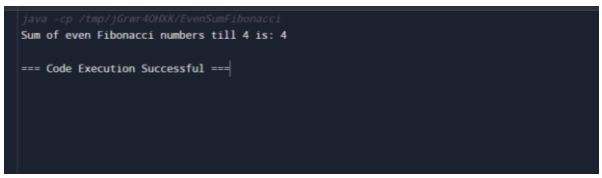
Sample Input: n = 4

Sample Output: 33

(N = 4, So here the fibonacci series will be produced from 0th term till 8th term:0, 1, 1, 2, 3, 5, 8, 13, 21

Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33)

```
public class EvenSumFibonacci {
public static void main(String[] args) {
int n = 4;
int sum = 0;
int a = 0;
int b = 1;
for (int i = 0; i <= n; i++) {
if (i % 2 == 0) {
sum += a;
}
int next = a + b;
a = b;
b = next;
}
System.out.println("Sum of even Fibonacci numbers till " + n + " is: " + sum);
}
}
```



5.Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50

N = 100

K = 7

Sample Output:

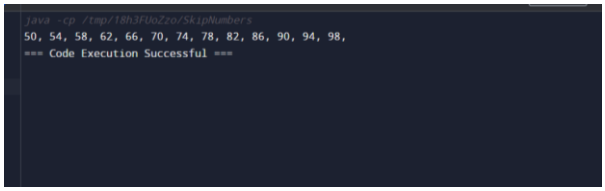
50, 58, 66, 74, …..

Test cases:

1. M = 15, N = 05, K = 02
2. .M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0 , N = 0 , K = 2
5. M = 200 , N = 200 , K = 50

```
public class SkipNumbers {
public static void main(String[] args) {
int M = 50;
int N = 100;
int K = 4;

for (int i = M; i <= N; i += K) {
System.out.print(i + ", ");
}
}
}
```



6.Write a program to print all the composite numbers between a and b?

Sample Input:

A = 12

B = 19

Sample Output

14, 15, 16, 18

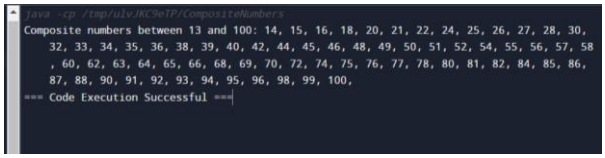
Test cases:

1. A = 11, B = 11
2. A = 20, B = 10
3. A = 0, B = 0
4. A = -5, B = 5
5. A = 7, B = -12

```
public class CompositeNumbers {
public static void main(String[] args) {
int A = 12;
int B = 19;

System.out.print("Composite numbers between " + A + " and " + B + ": ");
for (int i = A; i <= B; i++) {
if (isComposite(i)) {
System.out.print(i + ", ");
}
}
}
```

```
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;
}
}
```



7.Find the factorial of n?

Sample Input:

N = 4

Sample Output:

4 Factorial = 24

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = Q
5. N = 3A

```
import java.util.Scanner;
public class Factorial {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter a number to find its factorial: ");
int n = scanner.nextInt();
scanner.close();

if (n < 0) {
System.out.println("Factorial is not defined for negative numbers.");
} else {
int factorial = 1;
for (int i = 1; i <= n; i++) {
factorial *= i;
}
System.out.println(n + " Factorial = " + factorial);
}
}
}
```

```
java -cp /tmp/Y7UbVcWb7R/Factorial
Enter a number to find its factorial: 8
8 Factorial = 40320

=== Code Execution Successful ===
```

8.Find the year of the given date is leap year or not

Sample Input:

Enter Date: 04/11/1947

Sample Output:

Given year is Non Leap Year

Test cases:

1. 04/11/19.47
2. 11/15/1936
3. 31/45/1996
4. 64/09/1947
5. 00/00/2000

```
import java.util.Scanner;
public class LeapYearChecker {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
System.out.print("Enter Date: ");
String date = scanner.next();
String[] parts = date.split("/");
int year = Integer.parseInt(parts[2]);
if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
System.out.println("Given year is a Leap Year");
} else {
System.out.println("Given year is a Non Leap Year");
}
scanner.close();
}
}
```

```
java -cp /tmp/vE2uC1jXK1/LeapYearChecker
Enter Date: 20/11/2003
Given year is a Non Leap Year

=== Code Execution Successful ===
```

9.Find the number of factors for the given number

Sample Input:

Given number: 100

Sample Output:

Number of factors = 9

Test cases:

1. 343
2. 1080
3. -243
4. 101010
5. 0

```
public class NumberOfFactors {
```

```
public static void main(String[] args) {
int number = 100;
int count = 0;

for (int i = 1; i <= number; i++) {
if (number % i == 0) {
count++;
}
}

System.out.println("Number of factors = " + count);
}
}
```

```
Number of factors = 4

=== Code Execution Successful ===
```

10. Write a program to print the given number is Perfect number or not?

Sample Input:

Given Number: 6

Sample Output:

It’s a Perfect Number

Test cases:

1. 17
2. 26!
3. 143
4. 84.1
5. -963

```
public class PerfectNumber {
public static void main(String[] args) {
int givenNumber = 6;
if (isPerfectNumber(givenNumber)) {
System.out.println("It's a Perfect Number");
} else {
System.out.println("It's not a Perfect Number");
}
}

public static boolean isPerfectNumber(int number) {
if (number <= 0) {
return false;
}
int sum = 0;
for (int i = 1; i < number; i++) {
if (number % i == 0) {
sum += i;
}
}
```

```
return sum == number;
}
}
```