



Introduction to Computer Programming (CSE1001)

Assignment-5

Iterative Statements and Looping-II

Question No	Questions	Course Outcome
1.	<p>WAP to check whether a number is twisted prime or not.</p> <p>Twisted prime is a number if the number and its reverse both are prime then it is called twisted prime.</p> <p>Sample run 1: Enter a number: 97 97 is twisted prime. Sample run 2: 43 43 is not a twisted prime</p> <p>Sample run1: Enter a number: 97 97 is twisted prime.</p> <p>Sample run2: Enter a number: 43 43 is not a twisted prime.</p>	CO4
2.	<p>Amicable numbers are pair of numbers each of whose divisors are added to give the other number.</p> <p>Example: The smallest pair of amicable numbers is (220, 284). They are amicable because the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110, of which the sum is 284; and the proper divisors of 284 are 1, 2, 4, 71 and 142, of which the sum is 220.</p> <p>Note: 1 is included as a divisor but the numbers are not included as their own divisors.</p> <p>Write a Java Program that tests whether a given pair of numbers is amicable numbers or not.</p> <p>Sample run1: Enter first number: 220 Enter second number: 284 220 and 284 are amicable numbers.</p> <p>Sample run2: Enter first number: 220 Enter second number: 230 220 and 230 are not amicable numbers.</p>	CO4
3.	<p>A Strong number is a number in which the sum of the factorials of its digits is equal to the number itself.</p> <p>Example: 145 is a Strong number because: $1! = 1$ $4! = 24$ $5! = 120$ $Sum = 1 + 24 + 120 = 145$</p>	CO4

Write a **Java Program** that checks whether a given number is a **Strong number** or not.

Use **loops** and **conditional statements**.

Sample run 1:

Enter a number: 145
145 is a Strong number.

Sample run 2:

Enter a number: 123
123 is not a Strong number.

4. A divisor of a number is any **positive integer** that divides the number exactly, **including 1 and the number itself**. CO6

Write a Java program that accepts **two integers**, $n1$ and $n2$, and finds the number between **n1 and n2 (inclusive)** that has the **highest number of divisors**.

Your program should:

1. Read two integers: **n1** and **n2**
2. Check every integer from **n1 to n2**
3. Count all divisors of each number
4. Determine the number that has the **most divisors**.
5. If two numbers have **same divisor count**, the program selects the **larger number**.
6. Display:
 - o The number with the highest number of divisors
 - o The total number of divisors it has

Sample run 1:

Enter first number: 10
Enter second number: 20
The number with the most divisors is 18
Number of divisors: 6

Sample Run 2:

Enter first number: 1
Enter second number: 15
The number with the most divisors is 12
Number of divisors: 6

5. Write a **Java Program** to input a number **n**, and find the sum of the following series: $1 + (1+2) + (1+2+3) + \dots + (1+2+3+\dots+n)$ CO4

The program should display the final total sum of the entire series.

Sample run 1:

Enter the value of n: 4
The sum of the series is: 20

Sample run 2:

Enter the value of n: 6
The sum of the series is: 56

6.	<p>Write a Java Program that takes an integer n as input from the user and calculate the sum of the following series:</p> $\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2}$ <p>Assume sum as of double type.</p> <p>Sample run 1: Enter n: 5 Sum = 1.463611111111112</p> <p>Sample run 2: Enter n: 10 Sum = 1.5497677311665408</p>	CO4
7.	<p>Write a Java Program to display the following patterns using loop.</p> <p>(a)</p> <pre> * * * * * * * * * * * * * * * </pre> <p>(b)</p> <pre> 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 </pre> <p>(c)</p> <pre> 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 </pre> <p>(d)</p> <pre> A A B A B C A B C D A B C D E </pre>	CO4
8.	<p>Write a Java Program to display the following patterns using loop.</p> <p>(a)</p> <pre> \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ </pre> <p>(b)</p> <pre> \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ </pre> <p>(c)</p> <pre> \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ </pre> <p>(d)</p> <pre> \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ </pre>	CO4

9.	<p>Write the Java Program to find the sum of the following series.</p> $\frac{2}{1} - \frac{4}{3} + \frac{6}{5} - \frac{8}{7} + \dots + (-1)^{n+1} \frac{2n}{2n-1}$ <p>Sample run 1: Enter n: 5 Sum = 2.009523809523809</p> <p>Sample run 2: Enter n: 3 Sum = 1.8666666666666667</p>	CO5
10.	<p>Write a Java Program to display the following patterns using loop.</p> <pre> 5 4 3 2 * 5 4 3 * 1 5 4 * 2 1 5 * 3 2 1 * 4 3 2 1 </pre>	

HOME ASSIGNMENT

11.	<p>Write a Java Program that accepts two integers from the user (a first number and a second number) and displays all the prime numbers that lie between those two numbers (inclusive).</p> <p>Sample run: Enter the first number: 4 Enter the second number: 15 Prime numbers between 4 and 15 are: 5 7 11 13</p>	CO4
12.	<p>Write a Java program to calculate and display the factorial of all numbers between m and n (where m < n, m > 0, n > 0)</p> <p>Sample run: Enter the value of m: 2 Enter the value of n: 5 Factorial of 2 is: 2 Factorial of 3 is: 6 Factorial of 4 is: 24 Factorial of 5 is: 120</p>	CO4
13.	<p>Write a Java program to evaluate the function sin (x) as defined by the infinite series expansion.</p> $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ <p>The acceptable error for computation is 10^{-6}.</p>	CO6
14.	<p>Write a Java program to evaluate the function Cos (x) as defined by the infinite series expansion.</p> $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$ <p>The acceptable error for computation is 10^{-6}.</p>	CO6

15.

Given the sequence where the first three terms are:

- $a = 0$
- $b = 1$
- $c = 1$

Every subsequent term is the sum of the previous three terms.

Write a Java program to generate and display this sequence up to n terms, where $n > 3$.

CO5

Sample run:

```
Enter n (n > 3): 10  
0 1 1 2 4 7 13 24 44 81
```
