**Assignment 1. Name : Rushikesh A. Nigade**

## Date : 01.03.2022

**Submission date: 03.01.2022**

**Write Algorithm & Flowchart for following programs.**

1. Check the given number is EVEN or ODD. Step 1: Start.

Step 2: Read a number to N.

Step 3: Divide the number by 2 and store the remainder in R. Step 4: If R = O Then go to Step 6.

Step 5: Print “N is odd” go to step 7. Step 6: Print “N is even”

Step 7: Stop

1. Write a Java Program to find the Factorial of given number. Step 1: Start.

Step 2: number in a variable n. [We have to find factorial for thisnumber.] Step 3: Initialize variable factorial with 1.

Step 4: loop control variable i with 1.

Step 5: if i is less than or equal to n. If the condition is false, go to step 8. Step 6: factorial with i.

Step7: i. Go to step 5. Step 8: factorial.

Step 9: Stop.

1. Write the Factorial of a number using Recursion. Step 1: Start

Step 2: Read number n Step 3: Call factorial(n) Step 4: Print factorial f Step 5: Stop Factorial(n)

Step 1: If n==1 then return 1 Step 2: Else F=n\*factorial(n-1)

Step 3: Return f

1. Swap two numbers without using third variable approach. Step 1: START.

Step 2: ENTER x, y. Step3: PRINT x, y. Step 4: x= x+y.

Step 5: y= x – y.

Step 6: x =X – y.

Step 7: PRINT x, y.

Step 8: END.

1. How to check the given number is Positive or Negative in Java?

Step 1: Start

Step 2: Declare a variable.

Step3: Intialize the variable.

Step 4: Use the relational operator to check whether positive or negative.

Step 5: Display the result.

Step 6: Stop.

1. Write a Java Program to find whether given number is Leap year or NOT? Step 1: Start

Step 2: Take integer variable year Step 3:. Assign a value to the variable

Step 4: Check if the year is divisible by 4 but not 100, DISPLAY “leap year”

Step 5: Check if the year is divisible by 400, DISPLAY “leap year”

Step 6: Otherwise, DISPLAY “not leap year”

Step 7: Stop.

1. Write a Java Program to Print 1 To 10 Without Using Loop. Step 1: Start

Step 2: Take a number n

Step 3: call function printno(n) Step 4: Stop

Printno(n)

Step 1: if n>0 then return printno(n-1) Step 2: Display result

Step 3: Stop.

1. Write a Java Program to print the digits of a Given Number.

Step 1: Start

Step 2: create an array of strings to store digit to word mapping. Step 3: create another array digits[] to store individual digits of n.

Step 4: traverse digits of n and store them in digits[]. Note that standard traversal by repeated storing n%10 and doing n = n/10, traverses

digits in reverse order

Step 5: Traverse the digits array from end to beginning and print words

using the mapping created in step 1.

Step 6: Stop.

1. Write a Java Program to print all the Factors of the Given number. Step 1: Srart

Step 2: Input number from user and store it in variable num . Step 3: Run a loop from 1 to num , increment 1 in each iteration.

Step 4: for each iteration inside loop check current counter loop variable i is a factor of num or not.

Step 5: Stop.

1. Write a Java Program to find sum of the digits of a given number. Step 1: Start.

Step 2: Get number by user.

Step 3: Get the modulus/remainder of the number.

Step 4: sum the remainder of the number.

Step 5: Divide the number by 10.

Step 6: Repeat the step 2 while number is greater than 0. Step 7: Stop.

1. Write a Java Program to find the smallest of 3 numbers (a,b,c) Step 1: Start.

Step 2: Take three numbers in a, b, c. Check if a is less than b.

Step 3: If above condition is true, go to step 5, else go to step 7.

Step 4: Check if c is less than a.

Step 5: If above condition is true, c is the smallest, else a is the smallest. Go to step 9.

Step 6: Check if b is less than c.

Step 7: If above condition is true, b is the smallest, else c is the smallest. Step 8: Stop.

1. How to add two numbers without using the arithmetic operators in Java?

Step 1: Start

Step 2: Take two numbers from user a,b

Step 3: increment a and decrement b until b is equal to 0

Step 4: Print result

Step 5: Stop.

1. Write a java program to Reverse a given number.

Step 1: Start

Step 2: Intilize reverse=0.

Step 3: Read digit

Step 4: Check whether digit>0 then go to step 5 else go to step 9

Step 5: reverse =reverse \*10

step 6: reverse-reverse+digit

Step 7: digit -digit/10

step 8: Go to step 4

step 9: Print reverse.

step 10: Stop

1. Write a Java Program to find GCD of two given numbers.

Step 1: Start

Step 2: Declare variable n1, n2, gcd=1, i=1

Step 3: Input n1 and n2

Step 4: Repeat until i<=n1 and i<=n2

Step 5: If n1%i==0 && n2%i==0:

Step 6: gcd = i

Step 7: Print gcd

Step 8: Stop

1. Write a java program to LCM of TWO given number.

Step 1: If a = 0 or b = 0, then return with lcm(a, b) = 0, else go to step 2.

Step 2: Calculate absolute values of the two numbers.

Step 3: Initialize lcm as the higher of the two values computed in step 2.

Step 4: If lcm is divisible by the lower absolute value, then return.

Step 5: Increment lcm by the higher absolute value among the two and go to step 4.

Step 6: Stop

1. Write a java program to LCM of TWO given number using Prime Factors method.

Step 1: Start

Step 2: Perform the prime factorization of each number then write it in exponential form. Align the common prime factor base whenever possible.

Step 3: For the numbers with a common prime factor base, select the prime number that has the highest power. The prime factor with the highest power implies that it occurs the most in the entire list.

Step 4: If a distinct prime factor has NO matching prime factor base in the list, immediately include this factor with its exponent in the collection of numbers that you will multiply later.

Step 5: To determine the Least Common Multiple (LCM), multiply all the numbers that you have collected or gathered from steps 2 and 3.

Step 6: Stop

1. Check whether the Given Number is a Palindrome or NOT.

Step 1: Start

Step 2: Input the number.

Step 3: Find the reverse of the number.

Step 4: If the reverse of the number is equal to the number, then return true. Else, return false.

Step 5: Stop

1. Write a Java Program to print all the Prime Factors of the Given Number.

Step 1: Start

Step 2: While n is divisible by 2, print 2 and divide n by 2.

Step 3: After step 1, n must be odd. Now start a loop from I = 3 to the square root of n. While I divides n, print I, and divide n by i. After I fails to divide n, increment I by 2 and continue.

Step 4: If n is a prime number and is greater than 2, then n will not become 1 by the above two steps.

Step 5: print n if it is greater than 2.

Step 6: Stop

1. To print the following series EVEN number Series 2 4 6 8 10 12 14 16 .....

Step 1: Start

Step 2: Initialize the variable n and I to 1.

Step 3: while i<=n

Step 4: if i%2==0

Step 5: Print the number

Step 6: Increment value of i

Step 7: Stop

1. To print the following series ODD number Series 1 3 5 7 9 11 13 ....

Step 1: Start

Step 2: Assign i=0,n

Step 3: Repeat steps 4,5&6 until i=n reaches

Step 4: if i%2!=0 goto step 5

Step 5: Print i

Step 6: Compute i=i+1

Step 7: Stop

Start