```
//Check if the given number is EVEN or ODD.
1. Start
2. Enter the number
3. if (num%2 == 0)
4.     print number is even
5. else
6.     print number is odd
7. End
```

// Write a Java Program to find the Factorial of a given number.

1. Start
2. Enter the number
3. Declare the variable factorial and let i = 1
4. check if i<=num and if false go to step 7
5. factorial = factorial * i
6. update i++ and go to step 4
7. print factorial
8. End

// Find the Factorial of a number using Recursion.

1. Start
2. Enter the number n and declare a variable factorial
3. Create a function fact
4. Pass the number n in function fact as argument [fact(n)]
5. Now in function fact if(n == 0) return 1
6. else return n* fact(n-1)
7. Print factorial from function fact i.e. factorial = fact(n)
8. End

```
// Swap two numbers without using the third variable approach.
1. Start
2. Enter the value of x and y
3. x = x + y
4. y = x - y
5. x = x - y
6. print x and y
7. End
```

// How to check whether the given number is Positive or Negative in Java?

1. Start
2. Enter the num
3. if num>0
4. print number is positive
5. else if num<0
6. print number is negative
7. else print number is zero
8. End

// Write a Java Program to find whether a given number is Leap year or NOT.

```
1. Start
2.  Enter the year
3. if (year%400 == 0)
4.    print year is a leap year
5. else if (year%100 == 0)
6.    print year is not a leap year
7. else if  (year%4 == 0)
8.    print year is a leap year
9. else print year is not a leap year
10. End
```

// Write a Java Program to Print 1 To 10 Without Using Loop.

1. Start
2. Create a function printnum
3. Pass the value 1 in function printnum
4. In funtion printnum check if n<=10
5. if true then print the number n and pass the value n+1 in funtion printnum
6. else then go to step 7
7. End

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

1. Start
2. Enter the number n
3. Declare a variable digit .
4. Check the condition n>0 by using while loop, if false go to step 9
5. In loop calculate digit = n%10
6. print digit
7. calculate n=n/10
8. go to step 4
9. End

// Write a Java Program to print all the Factors of the Given number.
1. Start
2. Enter the number
3. Declare a variable i
4. check the condition i<= number by using for loop and if false go to step 7
5. in loop check if number%i==0 then print the value of i
6. increment the value of i by 1
7. End

// Write a Java Program to find the sum of the digits of a given number.

1. Start
2. Enter the number n
3. Declare a variable digit and sum.
4. Initialize the sum =0
5. Check the condition n>0 by using while loop, if false go to step 10
6. In loop calculate digit = n%10
7. Calculate sum = sum+digit
8. calculate n=n/10
9. go to step 5
10. Print sum
11. End

// Write a Java Program to find the smallest of 3 numbers (a,b,c)

1. Start
2. Input the value of a b c
3. If a<b && a<c
4. then print a is smallest
5. else if b<a && b<c
6. then print b is smallest
7. else
8. print c is smallest
9. End

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

1. Start
2. Enter the value of x and y
3. Declare a variable carry of int type
4. Check the condtion y!=0 by using while loop and if false go to step 9
5. In while loop calculate carry = x & y by using bitwise 'and' operator
6. Now calculate x = x^y by using bitwise 'XOR' operator
7. Again calculate y=carry<<1 by using left shift operator
8. go to step 4
9. print the value of x
10. End

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

1. Start
2. Enter the number n
3. Declare a variable remainder and reverse.
4. Initialize the reverse =0
5. Check the condition n>0 by using while loop, if false go to step 9
6. In loop calculate remainder = n%10
7. Calculate reverse= reverse*10+ remainder
8. calculate n=n/10
9. Print reverse
10. End

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

1. Start
2. Enter the value of x and y
3. Declare and initialize a variable gcd to 1
4. Run a for loop for x and y from 1 to max of x and y i.e. i<=x && i<=y
5. Check the condition if i divides both x and y completely and if false go to step 8
6. if divides completely then store the value of i in gcd
7. update the value of i by 1
8. print the value of gcd
9. End

// Write a java program to LCM of TWO given numbers.

1. Start
2. Enter the value of x and y
3. Declare the variables gcd and lcm and initialize gcd to 1
4. Run a for loop for x and y from 1 to max of x and y i.e. i<=x && i<=y
5. Check the condition if i divides both x and y completely and if false go to step 8
6. if divides completely then store the value of i in gcd
7. update the value of i by 1
8. Now calculate lcm = (x*y)/gcd
9. Print the value of lcm
10. End

// Write a java program to LCM of TWO given numbers using the Prime Factors method.

1. Start
2. Enter the two numbers num1 and num2
3. Declare the variables max_div and flag
4. set flag =1
5. let max_div variable holds the max number between num1 and num2 by using ternary operator
6. Use the while loop by giving condition flag=1
7. In loop check if the remainder comes 0 when dividing max_div by num1 and num2
8. if true then print the value of max_div as LCM and break the loop and go to step 10
9. else increment the value of max_div by 1 and go to step 6
10. End

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

1. Start
2. Enter the number n
3. Declare a variable remainder and reverse.
4. Initialize the reverse =0
5. Check the condition n>0 by using while loop, if false go to step 10
6. In loop calculate remainder = n%10
7. Calculate reverse= reverse*10+ remainder
8. calculate n=n/10
9. go to step 5
10. check if reverse == n
11. if true then print number is palindrome else print not palindrome
12. End

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

1. Start
2. Enter the number num
3. Run a for loop from 2 to num/2 and increment 1 in each iteration i.e for(i=2; i<=num/2;i++)
4. Inside loop, check if i is a factor of num or not
5. if i is a factor then check if it is a prime or not
6. if it is a prime then print the value of i
7. End

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

1. Start
2. let i=2
3. print i
4. update i = i+2 and go to step 3
5. End

```
// To print the following series ODD number Series 1 3 5 7 9 11 13...
1. Start
2. let i=1
3. print i
4. update i = i+2 and go to step 3
5. End
```