

### Test Set-A

Write a java method called vowelcount and digitcount which accepts a string as input and returns the number of vowels and digits in it. The method header is given below:

// Returns the number of vowels in the String.

**public static int vowelcount(String str)**

Example: vowelcount("soAlter45"), returns 4

// Returns the number of digits in the String.

**public static int digitcount(String str)**

For example: digitcount ("soAlter45") returns 2.

**Sample Run:**

Enter the String: soAlter45

Number of Vowels Present in given String : 4

Number of digits present in given String: 2

### Test Set-B

**A palindromic prime is a prime number and also palindromic.**

For example, 131 is a prime and also a palindromic prime, as are 313 and 757. Write a java program using methods (isPrime() and isPalindromic()) to display the first 100 palindromic prime numbers.

**public static boolean isPrime(int n)**

//Returns true if n is prime else returns false

**public static boolean isPalindromic(int n)**

//Returns true if n is palindromic number else returns false

Display 10 numbers per line, separated by exactly one space, as follow:

2 3 5 7 11 101 131 151 181 191

313 353 373 383 727 757 787 797 919 929

### Test Set-C

Given an array nums containing N distinct numbers in the range [0, N], return the only number in the range that is missing from the array. Write a method to find the Missing number in the Array. The method header is given as: **public static int MissingNumber(int[] nums)**

**Note: There are no duplicates in the list.**

**Sample Run:**

Enter the value of N : 9

Enter array elements: 9 6 4 2 3 5 7 0 1

The missing number is: 8

**Explanation:** N = 9 since there are 9 numbers, so all numbers are in the range [0,9]. 8 is the missing number in the range since it does not appear in nums.

### Test Set-D

Write a java program with method isArmstrong() to display the first 100 armstrong numbers. Display 10 numbers per line, separated by exactly one space.

**public static boolean isArmstrong(int n)**

//Returns true if n is an armstrong number else returns false

**Note:** A number is thought of as an Armstrong number if the sum of its own digits raised to the power number of digits gives the number itself. For example, 0, 1, 153, 370, 371, 407 are three-digit Armstrong numbers and, 1634, 8208, 9474 are four-digit Armstrong numbers and there are many more.