Test Set-A

Given an array of integer numbers and an integer target, print indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same index element twice. You can return the answer in any order.

Example:

Input: arr = {2,7,11,15}, target = 9

Output: [0,1]

Explanation: Because arr[0] + arr[1] == 9, we print [0, 1].

Input: arr = {3,2,4}, target = 6

Output: [1,2]

Test Set-B

Write a java program using a method isPrime() to find the sum of all prime numbers in the array, except the smallest prime number.

public static boolean isPrime(int n)

//Returns true if the n is prime else returns false

Example:

input = {10, 41, 18, 50, 43,31,29, 25,59, 96, 67} representing the given array. Then the expected output is 41+43+31+29+59+67=241, which is the sum of all prime numbers in the array except the smallest prime number 29.

Test Set-C

Write a java program to print the following pattern using user input.

Note: User input is always an odd number.

Example: N=7

A B C D
A B C
A B
A
A B
A B C
A B C D

Test Set-D

Write a java program that takes a string as input and calculate the weight of the string as per the rule mentioned below:

1. Weight of all alphabetic characters that appear in the string should be added.
2. The weight of all vowels that appear in the sting should either be ignored.
3. All non-alphabetic characters in the string should be ignored
4. Weight of each letter is its position in the English alphabet system, i.e. weight of a=1, weight of b=2, weight of c=3 and so on... weight of y=25, weight of z=26.
5. Weight of upper-case and lowercase letters should be taken as the same.

Example:

Weight of "Hello World!!" = 8+0+12+12+0+0+23+0+18+12+4+0+0=89