Find 0 with Farthest 1s in a Binary Array

Given a string (**seats**) of 1s and 0s, where **1** represents a filled seat and **0** represents an empty seat in a row. Find an empty seat with maximum distance from an occupied seat. Return the maximum distance.

Examples:

Input: *Seats = "1000101"*
Output*: 2*
Explanation*: Geek can take 3rd place and have a distance of 2 in left and 2 in right.*

Input: *Seats = "1000"*
Output: *3*
Explanation*: Geek can take the rightmost seat to have a distance of 3.*

Odd Numbers but not div by 3.

Write a Java program to print all elements of an integer array that are divisible by 3 but not divisible by 6.

Description:

You are given an array of integers. Your task is to iterate through the array and print only those elements that are divisible by 3 but **not divisible by 6**.

Example:

Input Array:
[3, 6, 9, 12, 15, 18, 21, 24, 27]

Output:
3 9 15 21 27

Explanation:

- 3, 9, 15, 21, and 27 are divisible by 3
- But they are **not divisible by 6**