Background:  
  
*There are 3 tasks here that pertains to C#, SQL, and JavaScript development.*

*Please don’t spend a lot of time with these. A couple of hours is ok.*

*You may submit your solution as a zipped file or in a Github or Gitlab repository.*

*Good luck!*

Task 1 – C#:  
  
Normally numbers are ordered 1, 2, 3, ..., 9, but in an alternative universe, the numbers are ordered as follows:

7, 9, 6, 4, 1, 3, 5, 8, 2

The task is to create a function called **GarbleSort** to sort a list of numbers with the values 1 to 9 (inclusive) according to the alternative ordering above.

Examples:

GarbleSort({ 1, 2, 3 }) = { 1, 3, 2 }

GarbleSort({ 5, 6, 3 }) = { 6, 3, 5 }

You can create a C# Console project or just a C# file.

## Task 2 - SQL:

In this problem, please provide just the SQL script file for the solution.

Amber's conglomerate corporation just acquired some new companies. Each of the companies follows this hierarchy:

Given the table schemas below, write a query to print the *company\_code*, *founder* name, total number of *lead* managers, total number of *senior* managers, total number of *managers*, and total number of *employees*. Order your output by ascending *company\_code*.

**Note:**

* The tables may contain duplicate records.
* The *company\_code* is string, so the sorting should not be **numeric**. For example, if the *company\_codes* are *C\_1*, *C\_2*, and *C\_10*, then the ascending *company\_codes* will be *C\_1*, *C\_10*, and *C\_2*.

**Input Format**

The following tables contain company data:

* Company: The company\_code is the code of the company and founder is the founder of the company.
* Lead\_Manager: The lead\_manager\_code is the code of the lead manager, and the company\_code is the code of the working company.  
   
* Senior\_Manager: The senior\_manager\_code is the code of the senior manager, the lead\_manager\_code is the code of its lead manager, and the company\_code is the code of the working company.  
   
* Manager: The manager\_code is the code of the manager, the senior\_manager\_code is the code of its senior manager, the lead\_manager\_code is the code of its lead manager, and the company\_code is the code of the working company.  
   
* Employee: The employee\_code is the code of the employee, the manager\_code is the code of its manager, the senior\_manager\_code is the code of its senior manager, the lead\_manager\_code is the code of its lead manager, and the company\_code is the code of the working company.  
   

**Sample Input**

* Company Table:  
   
* Lead\_Manager Table:  
   
* Senior\_Manager Table:  
   
* Manager Table:  
   
* Employee Table:  
   

Task 3 - JavaScript:

In this problem, please provide a Node.js project or just a single JS file.

Given an array of integers, find the one that appears an odd number of times.

There will always be only one integer that appears an odd number of times.

Examples:

[7] should return 7, because it occurs 1 time (which is odd).

[0] should return 0, because it occurs 1 time (which is odd).

[1,1,2] should return 2, because it occurs 1 time (which is odd).

[0,1,0,1,0] should return 0, because it occurs 3 times (which is odd).

[1,2,2,3,3,3,4,3,3,3,2,2,1] should return 4, because it appears 1 time (which is odd).