Single Number

Given a **non-empty** array of integers `nums`, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

```
Input: nums = [2,2,1]

Output: 1
```

Example 2:

```
Input: nums = [4,1,2,1,2]

Output: 4
```

Example 3:

```
Input: nums = [1]

Output: 1
```

```
public class Solution
{
        public int SingleNumber(int[] nums)
        {
                int res = 0;
                foreach(int number in nums)
                {
                        res ^= number;
                }
                return res;
        }
}
```

```
/*
        Bitwise XOR
        Bitwise XOR operator is represented by ^. It performs bitwise XOR operation on the corresponding bits of two operands.
                                If the corresponding bits are same, the result is 0. If the corresponding bits are different, the result is 1.

        For example:
        Let's take following array: [2, 1, 5, 1, 2]

        0010 = 2
        0001 = 1
        0101 = 5
        0010 = 2
        0001 = 1

        2 ^ 1
        0010
      + 0001
        -----
        0011 = 3

        3 ^ 5
        0011
      + 0101
        -----
        0110 = 6

        6 ^ 2
        0110
      + 0010
        -----
        0100 = 4

        4 ^ 1
        0100
      + 0001
        -----
        0101 = 5

        Solution - 0101 = 5
    */
```