FIND NUMBER OCCURRING ODD NUMBER OF TIMES IN AN ARRAY

Given an array of positive integers in which each number is repeated even number of times, except one which is repeated odd number of times.

Write an O(n) time algorithm to find the Number repeating odd number of times.

For Example, If the input array is:

```
{1, 2, 9, 2, 8, 1, 9, 2, 8, 1, 1}
```

Then Output should be 2, (all other integers are repeating even number of times in array)

Solution:

The solution to this problem uses the XOR operator. Result of XOR is as per the below truth table:

```
 A   B    A^B (A XOR B)
-- --   ---
0   0    0
0   1    1
1   0    1
1   1    0
```

XOR is ‘1‘, if the two bits are different and ‘0′ if they are same. Further, if we XOR any bit with zero, then the result is that bit itself

1 ^ 0 = 1

0 ^ 0 = 0

If we extend this to integers: *XOR of an unsigned int with itself is zero and XOR of an unsigned int with zero is the integer itself.*

```
A ^ A = 0

A ^ 0 = A
```

Further, If we XOR a number with itself even number of times, the result is zero. And If we XOR a number with itself odd number of times, the result it the number itself.

```
EVEN

----

   A^A^A^A = (A^A)^A^A = 0^A^A = A^A = 0

ODD

----

   A^A^A = (A^A)^A = 0^A = A
```

Algorithm:

XOR all the elements in the array. the numbers which are repeating even number of times will result to zero and the number repeating odd number of times will result in that number itself.

Hence we will be left with the number repeating odd number of times.

Code:

```
1    int getOddRepeatingNumber(int *arr, int n)
2    {
3        int result = 0;
4        for(int i=0; i < n; i++)
5            result ^= arr[i];
6
7        return result;
8    }
```

Note: *Result of an XOR operation is not defined if numbers are signed.*

Source: http://www.ritambhara.in/find-number-occurring-odd-number-of-times-in-an-array/