Find the odd int

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).

- array
- for every number it iterates through the give array, it checks if that number already exists in the new array, if not, then adds one, if it does, then removes one
- at the end of the array it should only leave the numbers that appear an odd number of times

my initial code was :

```
def find_it(seq):
    final_array = []
    for i in seq:
        if i in seq:
            final_array.remove(i)
        else:
            final_array.append(i)
    return final_array[0]
```

flawed as in i in seq will always be true, and therefore it will always remove(i)

```
def find_it(seq):
 counts = {}
 for i in seq:
  if i in counts:
   counts[i] += 1
  else:
   counts[i] = 1
 for key, value in counts.items():
  if value % 2 != 0:
   return key
```

this implements a dictionary in which the key (being index of the sequence) has a value (the count)

for every i in sequence that matches i in counts, the counter goes up by 1

and then the value is then quotient by 2, and if its not 0, then it is odd

A way to further improve this is to use a XOR operator

Exclusive Or

```
def find_it(seq):
result = 0
 for num in seq:
  result ^= num
return result
```

XOR(?)

a number XOR with itself cancels out

a number XOR with 0 will become the number itself

XOR is commutative and associative

in which order does not matter and the way you group them together operationally does not matter

This means at the end of the list, the only number that remains is the number in which it was odd

however it is limited in regards to it being able to identify XOR results rather then individual numbers specifically