

## **Xoring**

We are given an array containing  $n$  numbers. All the numbers are present twice except for one number which is only present once. Find the unique number without taking any extra spaces and in linear time.

### **Input Format**

An integer vector is passed in the function

### **Constraints**

$n < 10^5$

### **Output Format**

return a single integer containing the unique number

### **Sample Input**

{ 1, 2, 3, 1, 4, 2, 3 }

### **Sample Output**

4

**Solution:** xoring.cpp

## **Modulo Exponentiation**

Given three numbers  $x$ ,  $y$  and  $mod$ , compute  $x$  raised to power  $y$  modulo  $mod$  i.e.  $(x^y) \% mod$

### **Input Format**

Three integers  $x$ ,  $y$  and  $mod$  are passed.

### **Output Format**

Return an integer

### **Sample Input**

$x = 12, y = 25, mod = 10007$

### **Sample Output**

9603

**Solution:** moduloExp.cpp

## **Earth Levels!**

The Planet Earth is under a threat from the aliens of the outer space and your task is to defeat an enemy who is N steps above you (assume yourself to be at ground level i.e. at the 0th level). You can take jumps in power of 2. In order to defeat the enemy as quickly as possible you have to reach the Nth step in minimum moves possible.

### **Input Format**

In the function an integer is passed.

### **Output Format**

Return an integer representing minimum jumps.

### **Sample Input**

7

### **Sample Output**

3

### **Explanation**

0 -> 4 -> 6 -> 7

**Solution:** earthLevels.cpp

## **Subset Sum Queries**

Given an array and N number of queries, where in each query you have to check whether a subset whose sum is equal to given number exists in the array or not.

### **Input Format**

In the function two vectors num and query are passed.

### **Output Format**

Return a bool vector of size N representing each query.

### **Sample Input**

num[] = {1, 2, 3} query[] = {5, 3, 8}

**Sample Output**

{Yes, Yes, No}

**Solution:** subsetSum.cpp