Introduction to Bit Manipulation

```java
import java.io.*;
import java.util.*;

public class Main {

  public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int i = scn.nextInt();
    int j = scn.nextInt();
    int k = scn.nextInt();
    int m = scn.nextInt();

    //write your code here
    int onmask = (1 << i);
    int offmask = ~(1 << j);
    int tmask = (1 << k);
    int cmask = (1 << m);

    System.out.println(n | onmask);
    System.out.println(n & offmask);
    System.out.println(n ^ tmask);
    System.out.println((n & cmask) == 0? false: true);
  }

}
```

Right Most Set Bit (RSB) Mask

```java
import java.io.*;
import java.util.*;

public class Main {

  public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    //write your code here
    int rsbm = n & -n;

    System.out.println(Integer.toBinaryString(rsbm));
  }

}
```

Kernighan's Algorithm | Count Set Bits in an Integer

```java
import java.io.*;
import java.util.*;

public class Main {

  public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();

    //write your code here
    int counter = 0;
    while(n != 0){
        int rsbm = n & -n;
        n -= rsbm;
        counter++;
    }

    System.out.println(counter);
  }

}
```

Josephus Problem Algorithm using Bit Manipulation |

```java
public class Main {

  public static int powerof2(int n){
    int i = 1;

    while(i * 2 <= n){
        i = i * 2;
    }

    return i;
  }

  public static int solution(int n){
    int hp2 = powerof2(n);
    int l = n - hp2;
    return 2 * l + 1;
  }
  public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    System.out.println(solution(n));
  }
```

Gray Code Explained using Recursion and Backtracking | Leetcode#89 Solution in JAVA

```java
public class Main {

  public static ArrayList<String> solution(int n) {
        if(n == 1){
            ArrayList<String> bres = new ArrayList<>();
            bres.add("0");
            bres.add("1");
            return bres;
        }

        ArrayList<String> rres = solution(n - 1);
        ArrayList<String> mres = new ArrayList<>();
        for(int i = 0; i < rres.size(); i++){
            String rstr = rres.get(i);
            mres.add("0" + rstr);
        }

        for(int i = rres.size() - 1; i >= 0; i--){
            String rstr = rres.get(i);
            mres.add("1" + rstr);
        }
```

Minimum Number of Developers

```java
public class Main {

    static ArrayList<Integer> sol = new ArrayList<>();

    public static void solution(int[] people, int nskills, int cp, ArrayList<Integer> onesol, int smask) {
        if(cp == people.length){
            if(smask == ((1 << nskills) - 1)){
                if(sol.size() == 0 || onesol.size() < sol.size()){
                    sol = new ArrayList<>(onesol);
                }
            }

            return;
        }

        solution(people, nskills, cp + 1, onesol, smask); // no

        onesol.add(cp);
        solution(people, nskills, cp + 1, onesol, (smask | people[cp]))
        onesol.remove(onesol.size() - 1);
    }
```

```java
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        HashMap<String, Integer> smap = new HashMap<>();
        for (int i = 0; i < n; i++) {
            smap.put(scn.next(), i);
        }

        int np = scn.nextInt();
        int[] people = new int[np];
        for (int i = 0; i < np; i++) {
            int personSkills = scn.nextInt();
            for (int j = 0; j < personSkills; j++) {
                String skill = scn.next();
                int snum = smap.get(skill);
                people[i] = people[i] | (1 << snum);
            }
        }

        solution(people, n, 0, new ArrayList<>(), 0);
        System.out.println(sol);

    }
```

```java
import java.io.*;
import java.util.*;

public class Main {

    public static ArrayList<Integer> findNumOfValidWords(String[] words, String[] puzzles) {
        HashMap<Character, ArrayList<Integer>> map = new HashMap<>();
        for(int i = 0; i < 26; i++){
            map.put((char)('a' + i), new ArrayList<>());
        }
```

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    String[] words = new String[n];
    for(int i = 0 ; i < words.length; i++) {
        words[i] = scn.next();
    }
    int m = scn.nextInt();
    String[] puzzles = new String[m];
    for(int i = 0 ; i < m ;i++) {
        puzzles[i] = scn.next();
    }
    ArrayList<Integer> ans = findNumOfValidWords(words,puzzles);
    for(int i = 0; i < ans.size(); i++) {
        System.out.print(puzzles[i] + " -> " + ans.get(i));
    }
}
```

```java
for(String word: words){
    int mask = 0;
    for(char ch: word.toCharArray()){
        int bit = ch - 'a';
        mask = mask | ((1 << bit));
    }

    HashSet<Character> unique = new HashSet<>();
    for(char ch: word.toCharArray()){
        if(unique.contains(ch)){
            continue;
        }

        unique.add(ch);
        map.get(ch).add(mask);
    }
}
```

```java
ArrayList<Integer> res = new ArrayList<>();
for(String puzzle: puzzles){
    int pmask = 0;
    for(char ch: puzzle.toCharArray()){
        int bit = ch - 'a';
        pmask = pmask | ((1 << bit));
    }

    char fch = puzzle.charAt(0);
    ArrayList<Integer> wordsToCheck = map.get(fch);
    int count = 0;

    for(int wmask: wordsToCheck){
        if((wmask & pmask) == wmask){
            count++;
        }
    }

    res.add(count);
}

return res;
}
```

Find element that appears once while all other elements appear twice | XOR Operator Implementation

```
public class Main {

  public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for(int i = 0 ; i < n; i++){
      arr[i] = scn.nextInt();
    }
    //write your code here
   int uni = 0;
   for(int val : arr){
       uni = uni ^ val;
   }

   System.out.println(uni);
  }

}
```

All Repeating Except Two | Two Unique Rest Twice |

```
public static void solution(int[] arr){
  int xxory = 0;

  for(int val: arr){
      xxory = xxory ^ val;
  }

  int rsbm = xxory & -xxory;

  int x = 0;
  int y = 0;

  for(int val: arr){
      if((val & rsbm) == 0){
          x = x ^ val;
      } else {
          y = y ^ val;
      }
  }


  if(x < y){
      System.out.println(x);
      System.out.println(y);
  } else {
      System.out.println(y);
      System.out.println(x);
  }
}
```

```java
public static void main(String[] args){
  Scanner scn = new Scanner(System.in);
  int n = scn.nextInt();
  int[] arr = new int[n];
  for(int i = 0 ; i < n; i++){
    arr[i] = scn.nextInt();
  }
  solution(arr);
}

public static void solution(int[] arr){
 //write your code here

 int xor = 0;
 for(int i = 0; i < arr.length; i++){
     xor ^= arr[i];
 }

 for(int i = 1; i <= arr.length; i++){
     xor ^= i;
 }
```

```java
 int rsb = xor & -xor;

 int x = 0;
 int y = 0;

 for(int val: arr){
     if((val & rsb) == 0){
         x = x ^ val;
     } else {
         y = y ^ val;
     }
 }

 for(int i = 1; i <= arr.length; i++){
     if((i & rsb) == 0){
         x = x ^ i;
     } else {
         y = y ^ i;
     }
 }

 for(int val : arr){
     if(val == x){
         System.out.println("Missing Number -> " + y);
         System.out.println("Repeating Number -> " + x);
         break;
     } else if(val == y){
         System.out.println("Missing Number -> " + x);
         System.out.println("Repeating Number -> " + y);
         break;
     }
```


Find Duplicate Number and Missing Number from 1 to N | One Duplicate One Missing | Bit Manipulation