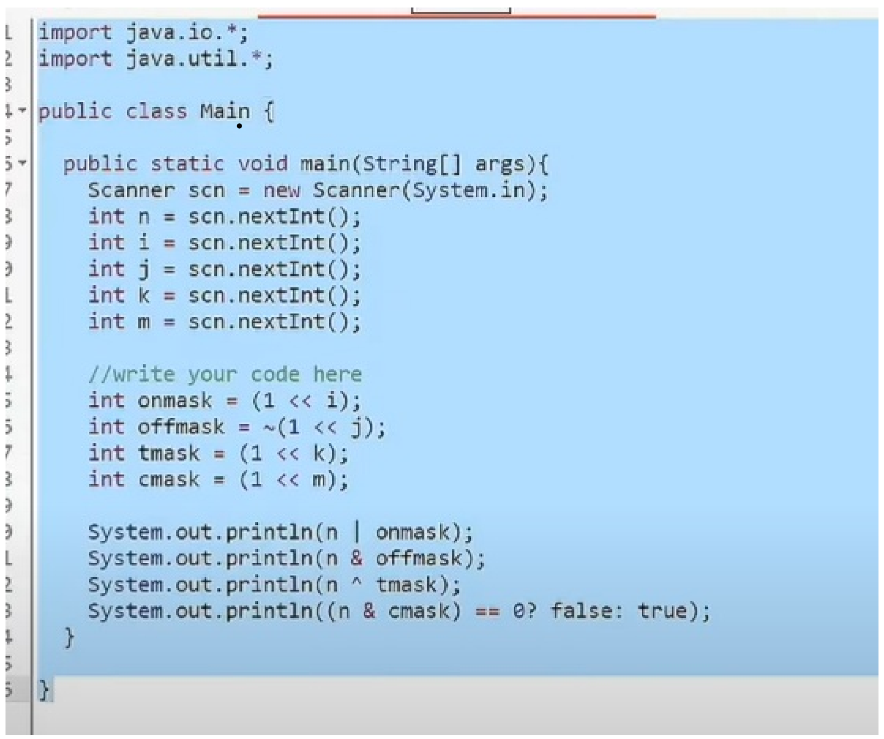
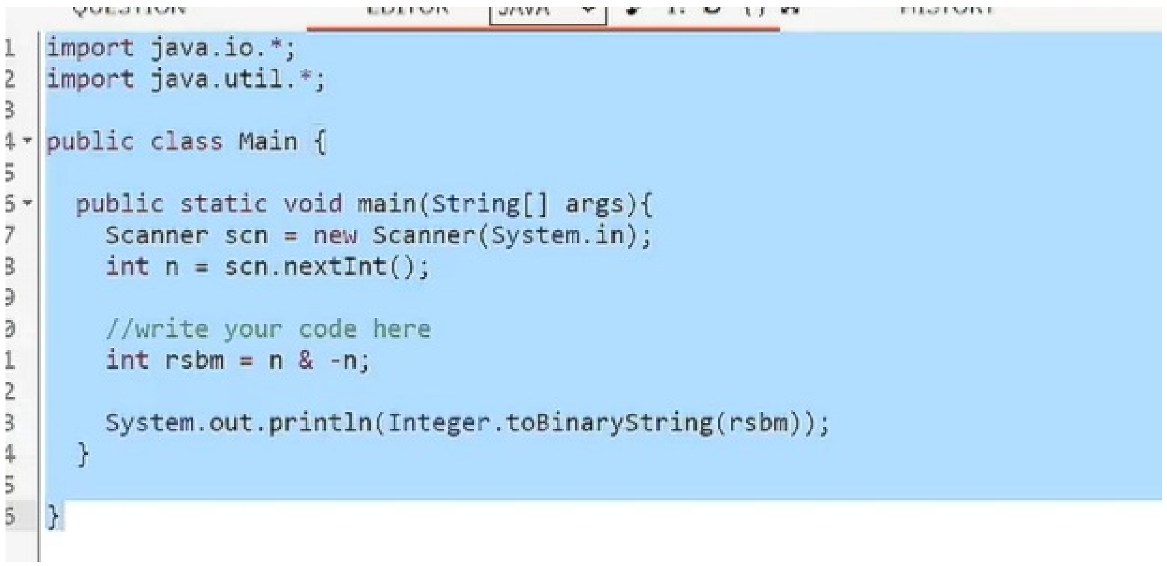
Introduction to Bit Manipulation



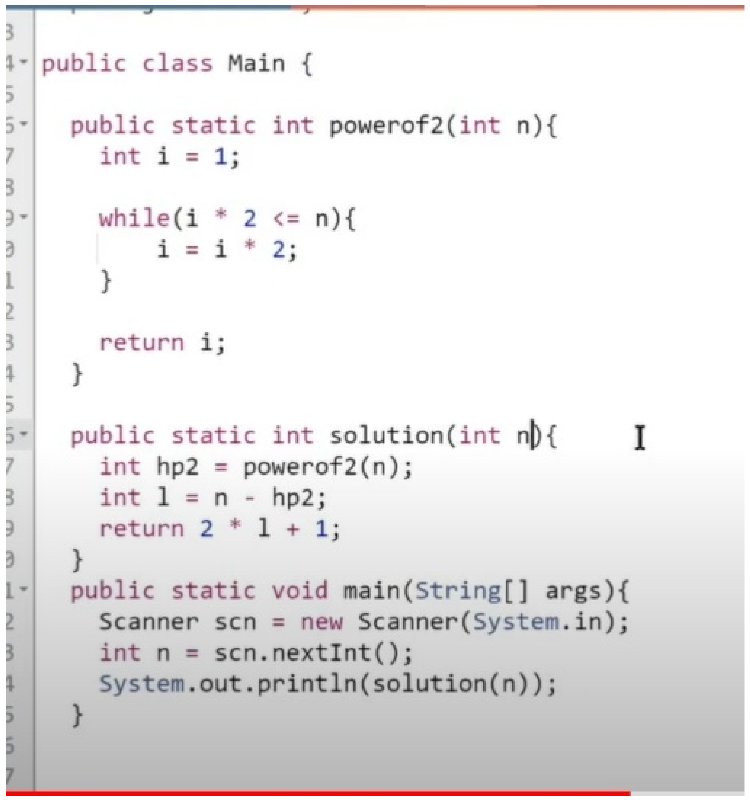
```java
import java.io.*;
import java.util.*;

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



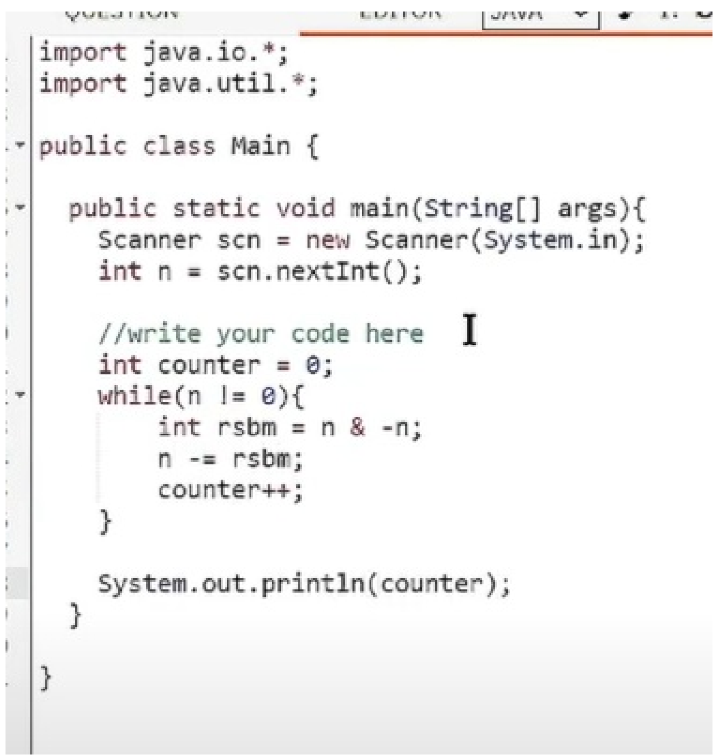
```java
import java.io.*;
import java.util.*;

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));
  }

}
```

Josephus Problem Algorithm using Bit Manipulation |



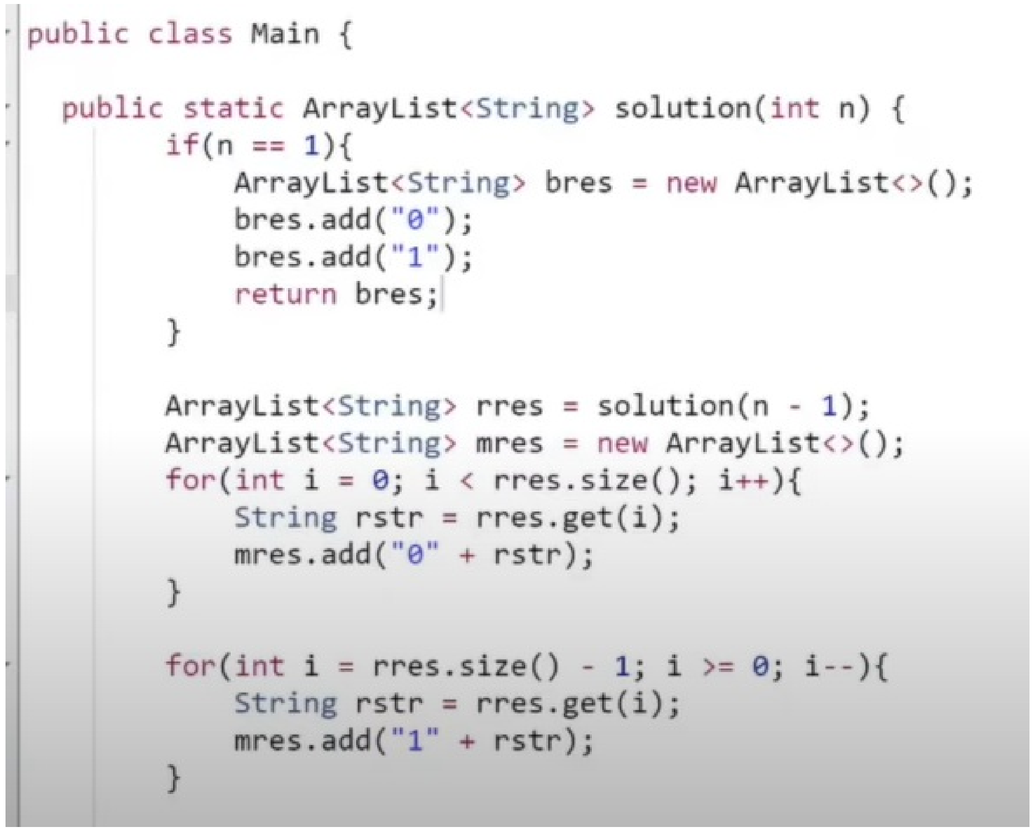
```java
public class Main {

  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));
  }
```

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);
  }

}
```

Gray Code Explained using Recursion and Backtracking | Leetcode#89 Solution in 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);
        }
```