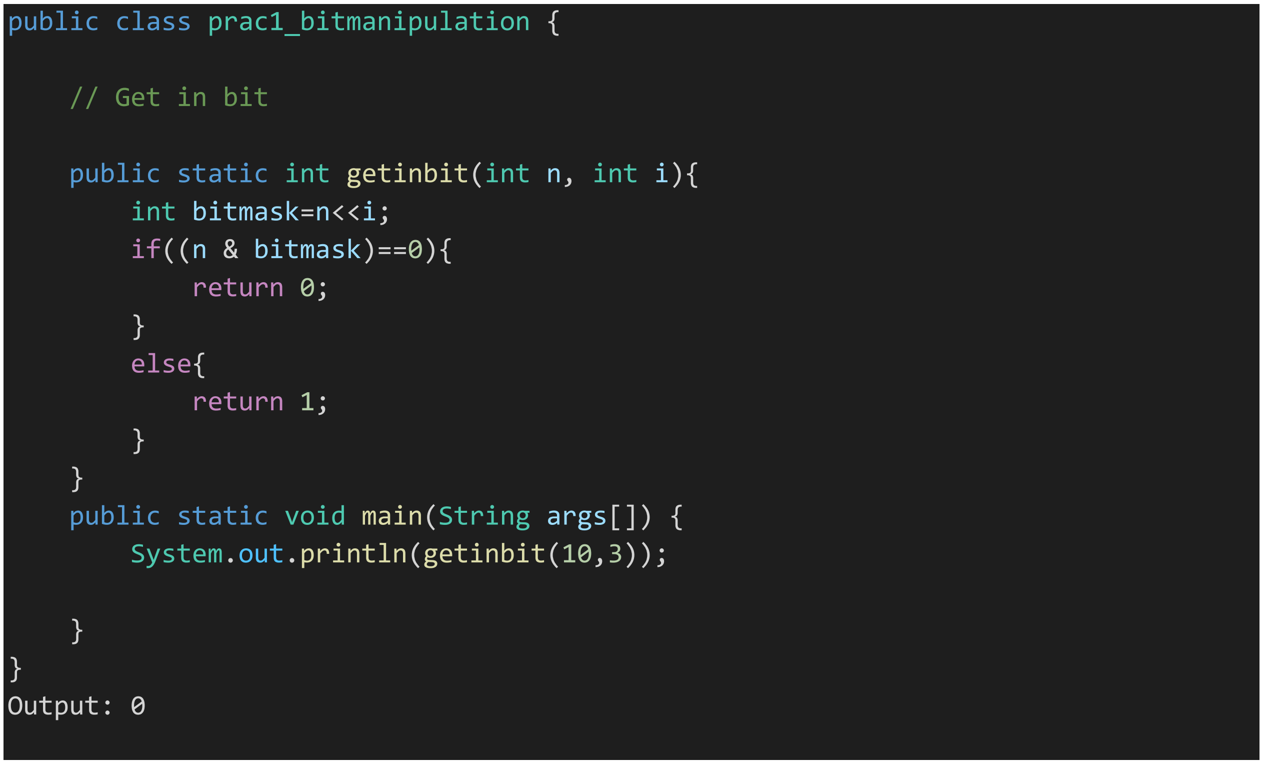
BIT MANIPULATION

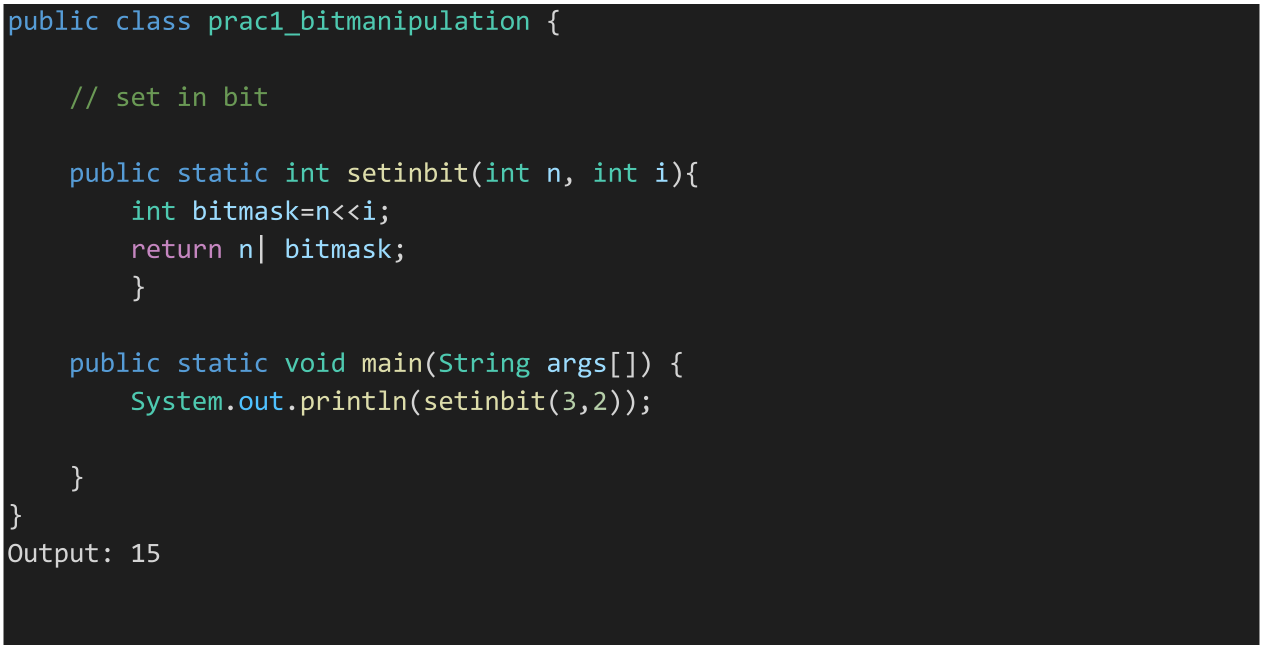
```
public class prac1_bitmanipulation {

    // bitwise and operation
    public static void main(String args[]) {
        System.out.println(5&6);

Output: 4


    // bitwise or operation
        System.out.println(5|6);


Output: 7


    // bitwise XOR operation

        System.out.println(5^6);

   // bitwise one's complement operation

        System.out.println(~0);

Output :-1

    // bitwise   left shift operation

        System.out.println(1<<6);

Output : 64

       // bitwise   left shift operation

Output : 3
```

Get in bit:



```
public class prac1_bitmanipulation {

    // Get in bit

    public static int getinbit(int n, int i){
        int bitmask=n<<i;
        if((n & bitmask)==0){
            return 0;
        }
        else{
            return 1;
        }
    }
    public static void main(String args[]) {
        System.out.println(getinbit(10,3));

    }
}
Output: 0
```

Set in bit :



```
public class prac1_bitmanipulation {

    // set in bit

    public static int setinbit(int n, int i){
        int bitmask=n<<i;
        return n| bitmask;
        }

    public static void main(String args[]) {
        System.out.println(setinbit(3,2));

    }
}
Output: 15
```

Clear ith bit:

```
public class prac1_bitmanipulation {

    // clear i tn bit

    public static int clearinbit(int n, int i){
        int bitmask=~(1<<i);
        return n& bitmask;
        }

    public static void main(String args[]) {
        System.out.println(clearinbit(10,1));

    }
}
Output: 15
```

Update ith bit:

```
public class prac1_bitmanipulation {
        // clear in bit

    public static int clearbit(int n,int i){
        int bitmask=~(1<<i);
        return n&bitmask;}

    // set in bit

    public static int setinbit(int n, int i){
        int bitmask=n<<i;
        return n| bitmask;
        }

 public static int updateithbit(int n, int i, int newbit)
 {
    if(newbit==0){
        return clearbit(n, i);

    }
    else{
        return setinbit(n, i);
    }
 }
    public static void main(String args[]) {
        System.out.println(updateithbit(2,2,1));

    }
}
Output: 10

```

Clear last bit:

```
public class clearlastbit {
    public static int clearlastbit(int n, int i){
        int bitmask = (~0)<<i;
        return n& bitmask;
    }
    public static void main(String args[]){
        System.out.println(clearlastbit(15,2));
    }

}
Output: 12
```

Clear range of bits:

```
public class clearrangeofbits {

    public static int clearrangeofbits(int n,int i,int j){
        int a= (~0)<<(j+1);
        int b= (1<<i)-1;
        int bitmask = a | b;
        return n& bitmask;
    }

    public static void main(String args[]){
        System.out.println(clearrangeofbits(10,2,4));
    }

}
Output: 2
```

Check a number is a power of 2 or not

(ex: 4 =2^2, 8=2^3, 32= 2^5 …..)

```
public class powerof2ornot {
    public static boolean ispowerof2(int n){
        return (n & (n-1))==0;


    }
    public static void main(String args[]){
        System.out.println(ispowerof2(16));

    }

}
Output: true
```

Count setbits(no of 1’s) in a number:

```
public class countnoofsetbitsinnumber {
  public static int countbits(int n ){
    int count=0;
    while(n>0){
        if((n & 1)!=0){
            count++;
        }
        n=n>>1;
    }
    return count;
  }


    public static void main(String args[]){
        System.out.println(countbits(10));

    }

}
Output: 2
```

Fast exponentiation

```
public class fastexponentioation {
    public static int fe(int a,int n){
        int ans=1;
        while(n>0){
            if((n &1)!=0){
                ans=ans*a;
            }
            a=a*a;
            n=n>>1;
        }
        return ans;
    }

    public static void main(String args[]){
        System.out.println(fe(5,3));
    }

}
Output: 125
```

Swaping 2 numbers without third variable

```
public class swaping2numbrswithout3rdvariable {
    public static void main (String args[]) {
        int x=4;
        int y=5;
        System.out.println("beforeswapping "+ x+" "+ y);
        x=x^y;// x xor=ans1
        y=x^y;// y xor =ans2

        x=x^y;//ans1 xor ans2
        System.out.println("afterswapping "+ x+" "+ y);


    }


}
```

```
Output:
beforeswapping 4 5
afterswapping 5 4
```

add number to integer by bits

```
public class addingintegertobit {
    public static void main(String args[]){
        int a=6;
        System.out.println(a+" + "+1+" = "+-~a);
        int b=-4;
        System.out.println(b+" + "+1+" = "+-~b);
        int c=0;
        System.out.println(c+" + "+1+" = "+-~c);
    }

}
Output:
6 + 1 = 7
-4 + 1 = -3
0 + 1 = 1
```

Uppercase to lowercase using bits

```
public class uppercasetolowecasebybits {
    public static void main(String args[]){
        for(char ch='A';ch<='Z';ch++){
            System.out.print((char)(ch |' '));
        }
    }

}
Output:
abcdefghijklmnopqrstuvwxyz
```

book for exploring Bit-manipulation

https://graphics.stanford.edu/~seander/bithacks.html