

BIT MANIPULATION

1. Add 1 to an Integer

```
import java.util.Scanner;
public class add1toAnInt {
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter any Integer: ");
        int num = sc.nextInt();

        // ~num = num + 1;
        System.out.println(num + " + " + 1 + " is: " + ~num);
    }
}
```

2. Binary Operators

```
import java.util.Scanner;
public class binaryOperators {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 1st Number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter 2nd Number: ");
        int num2 = sc.nextInt();

        // Binary Operators
        System.out.println("Binary AND: " + (num1 & num2));
        System.out.println("Binary OR: " + (num1 | num2));
        System.out.println("Binary XOR: " + (num1 ^ num2));
        System.out.println("Binary One's Complement: " + (~num2));
        System.out.println("Binary left shift: " + (num1 << num2));
        System.out.println("Binary right shift: " + (num1 >> num2));
    }
}
```

3. Clear ith Bit

```
import java.util.Scanner;
public class ClearithBit {
    public static int clear_i_bit (int n, int i) {
        int bitmask = ~(1<<i);
        return n & bitmask;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter value of i: ");
        int i = sc.nextInt();
        System.out.println("Clear ith bit of value is: " + clear_i_bit(num, i));
    }
}
```

4. Clear last ith Bit

```
import java.util.Scanner;

public class clearLastIthBit {

    public static int clear_last_ith_bit (int n, int i) {
        int bitMask = (~0)<<i;
        return n & bitMask;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter value of i: ");
        int i = sc.nextInt();
        System.out.println("After clearing last ith bit: " + clear_last_ith_bit(num, i));
    }
}
```

5. Clear Bits in Range

```
import java.util.Scanner;

public class clearRangeBits {

    public static int clear_range_bits (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) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter range i: ");
        int i = sc.nextInt();
        System.out.print("Enter range j: ");
        int j = sc.nextInt();
        System.out.println("After clearing bits in range: " + clear_range_bits(num, i, j));
    }
}
```

6. Count Set Bits

```
import java.util.Scanner;

public class countSetBits {

    public static int count_set_bits (int num) {
        int count = 0;
        while (num > 0) {
            if ((num & 1) != 0) { // Check LSB
                count++;
            }
            num = num >> 1;
        }
        return count;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.println("Set bit in a " + num + " is: " + count_set_bits(num));
    }
}
```

7. Fast Exponentiation

```
import java.util.Scanner;

public class fastExponentiation {

    public static int fast_expo (int base, int pow) {
        int ans = 1;
        while (pow > 0) {
            if ((pow & 1) != 0) {
                ans = ans * base;
            }
            base = base * base;
            pow = pow >> 1;
        }
        return ans;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Base: ");
        int base = sc.nextInt();
        System.out.print("Enter the Power: ");
        int pow = sc.nextInt();
        System.out.println("Exponentiation: " + fast_expo(base, pow));
    }
}
```

8. Get ith Bit

```
import java.util.Scanner;

public class GetIthBit {

    public static int get_ith_bit (int n, int i) {
        int bitmask = 1 << i;
        if (((n & bitmask) == 0)) {
            return 0;
        } else {
            return 1;
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter ith value: ");
        int i = sc.nextInt();
        System.out.println(i + " bit of " + num + " is: " + get_ith_bit(num, i));
    }
}
```

9. Number is Power of 2 or Not

```
import java.util.Scanner;

public class numIsPowof2ornot {

    public static boolean isPowerOfTwo (int num) {
        return (num & (num-1)) == 0;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.println(num + " is Power of 2: " + isPowerOfTwo(num));
    }
}
```

10. Uppercase to Lowercase

```
public class uppercaseToLowercase {

    public static void main(String[] args) {

        for (char ch = 'A'; ch <= 'Z'; ch++) {
            System.out.print((char)(ch | ' '));
        }
    }
}
```

11. Odd or Even

```
import java.util.Scanner;

public class oddOrEven {

    public static void odd_even(int num) {
        int bitmask = 1;
        if ((num & bitmask) == 0) {
            System.out.println("It's an Even Number.");
        } else {
            System.out.println("It's an Odd Number.");
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter any Number: ");
        int num = sc.nextInt();
        odd_even(num);
    }
}
```

12. Set ith Bit

```
import java.util.Scanner;

public class SetithBit {

    public static int set_ith_bit (int n, int i) {
        int bitmask = 1<<i;
        return n | bitmask;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter value of i: ");
        int i = sc.nextInt();
        System.out.println("Set ith Bit of the value is: " + set_ith_bit(num, i));
    }
}
```

13. Swapping of two without using third variable

```
import java.util.Scanner;
public class swapOfTwo {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter 1st number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter 2nd number: ");
        int num2 = sc.nextInt();
        System.out.println("Before Swapping: x = "+num1+", y = "+num2);

        num1 = num1 ^ num2;
        num2 = num1 ^ num2;
        num1 = num1 ^ num2;

        System.out.println("After Swapping: x = "+num1+", y = "+num2);
    }
}
```

14. Update the ith Bit

```
import java.util.Scanner;
public class UpdateithBit {

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

    public static int set_ith_bit(int n, int i) {
        int bitmask = 1 << i;
        return n | bitmask;
    }

    public static int update_ith_bit (int n, int i, int newBit) {
        // if (newBit == 0) {
        //     return clear_ith_bit(n, i);
        // } else {
        //     return set_ith_bit(n, i);
        // }

        n = clear_ith_bit(n, i);
        int bitmask = newBit << i;
        return n | bitmask;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter Number: ");
        int num = sc.nextInt();
        System.out.print("Enter value of i: ");
        int i = sc.nextInt();
        System.out.print("Enter newBit (0/1): ");
        int newBit = sc.nextInt();
        System.out.println("Updated ith bit of value is: " + update_ith_bit(num, i, newBit));
    }
}
```