Problem No. 01

Zero Converter:

You are given a number n. The number n can be negative or positive. If n is negative, print numbers from n to 0 by adding 1 to n in the neg function. If positive, print numbers from n-1 to 0 by subtracting 1 from n in the pos function.

Note:- You don't have to return anything, you just have to print the array.

Example 1:	Example 2:	Example 3:
Input:	Input:	Input:
n = 0	n = 4	n = -3
Output:	Output:	Output:
already Zero	3 2 1 0	-3 -2 -1 0

Your Task:
This is a function problem. You need to take input of test cases. Just complete the functions pos() and neg().

Constraints:
$-10^3 <= x <= 10^3$

Sol:

```
import java.util.Scanner;


public class Main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
```

```
    int t = sc.nextInt();
    while (t-- > 0) {
        int n = sc.nextInt();
        if (n == 0) {
            System.out.println("already Zero");
        } else if (n > 0) {
            for (int i = n - 1; i >= 0; i--) {
                System.out.print(i + " ");
            }
            System.out.println(0);
        } else {
            for (int i = n; i <= 0; i++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
    }
}

public static void pos(int n) {
    for (int i = n - 1; i >= 0; i--) {
```

```
            System.out.print(i + " ");
        }
        System.out.println(0);
    }

    public static void neg(int n) {
        for (int i = n; i <= 0; i++) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

Problem No.2

Write a simple ATM Program to display the following options and perform the user-selective operation until the user exits from the program.

- Initial account pin is 1234 and the balance of the user is 0.
- Read the PIN from the user and if it matches then print and execute the following operations as per user selection.

1. Deposit
2. Withdraw
3. Balance enquiry

4. PIN change
5. Exit
    - Return the total available balance after the deposit.
    - Return the total available balance after the withdrawal.
    - Return the total available balance after the Balance check.
    - Return “PIN Change” after the pin change operation.

Transaction(1) : input: 5000 output:5000

Sol:

```
import java.util.Scanner;



public class ATM {

    public static void main(String[] args) {

        int pin = 1234;

        int balance = 0;

        Scanner scanner = new Scanner(System.in);

        while (true) {

            System.out.println("Enter your PIN:");

            int inputPin = scanner.nextInt();
```

```
            if (inputPin == pin) {
                System.out.println("1. Deposit");
                System.out.println("2. Withdraw");
                System.out.println("3. Balance enquiry");
                System.out.println("4. PIN change");
                System.out.println("5. Exit");
                System.out.println("Choose the operation you want to perform:");
                int choice = scanner.nextInt();
                switch (choice) {
                    case 1:
                        System.out.println("Enter the amount to deposit:");
                        int depositAmount = scanner.nextInt();
                        balance += depositAmount;
                        System.out.println("Transaction successful. Total available
balance: " + balance);
```

```
                    break;
                case 2:
                    System.out.println("Enter the amount to withdraw:");
                    int withdrawAmount = scanner.nextInt();
                    if (withdrawAmount > balance) {
                        System.out.println("Insufficient balance.");
                    } else {
                        balance -= withdrawAmount;
                        System.out.println("Transaction successful. Total
available balance: " + balance);
                    }
                    break;
                case 3:
                    System.out.println("Total available balance: " + balance);
                    break;
```

```
            case 4:
                System.out.println("Enter your new PIN:");
                int newPin = scanner.nextInt();
                pin = newPin;
                System.out.println("PIN change successful.");
                break;
            case 5:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice.");
                break;
        }
    } else {
        System.out.println("Invalid PIN.");
```

```
        }
      }
    }
}
```

Problem No.3

Given two int values a and b. Check a and b are primes are not. If a and b are primes then check if the sum of a and b is a prime or not. If the prime returns true. Else return false.

SumPrime(2,3) —> true.

SumPrime(7,11) —> true.

SumPrime(3,7) —> false

SumPrime(8,11) → false

Sol:

```
public static boolean isPrime(int n) {
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i <= Math.sqrt(n); i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

public static boolean SumPrime(int a, int b) {
    if (isPrime(a) && isPrime(b)) {
        int sum = a + b;
        if (isPrime(sum)) {
            return true;
        }
    }
    return false;
}
```

```
public static boolean isPrime(int n) {
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i <= Math.sqrt(n); i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

public static boolean SumPrime(int a, int b) {
    if (isPrime(a) && isPrime(b)) {
        int sum = a + b;
        if (isPrime(sum)) {
            return true;
        }
    }
    return false;
```

}

Problem No.5

We use the integers a, b, and n to create the following series:

(a+2^0 *b), (a+2^0 *b+2^1*b), (a+2^0 *b+2^1*b +2^2*b), …… (a+2^0 *b+2^1*b+2^2*b+.... + 2^n-1 * b)

You are given q queries in the form of a, b, and n. For each query, print the series corresponding to the given a, b, and n values as a single line of space-separated integers.

Input Format: The first line contains an integer, q denoting the number of queries.
Each line i of the q subsequent lines contains three space-separated integers describing the respective ai, bi, ni, and values for that query.

Constraint:

0<=q<=500

0<=a,b<=50

0<=n<=15

Output Format

For each query, print the corresponding series on a new line. Each series must be printed in order as a single line of space-separated integers.

Sample Input

2

0 2 10

5 3 5

Sample Output

2 6 14 30 62 126 254 510 1022 2046

8 14 26 50 98