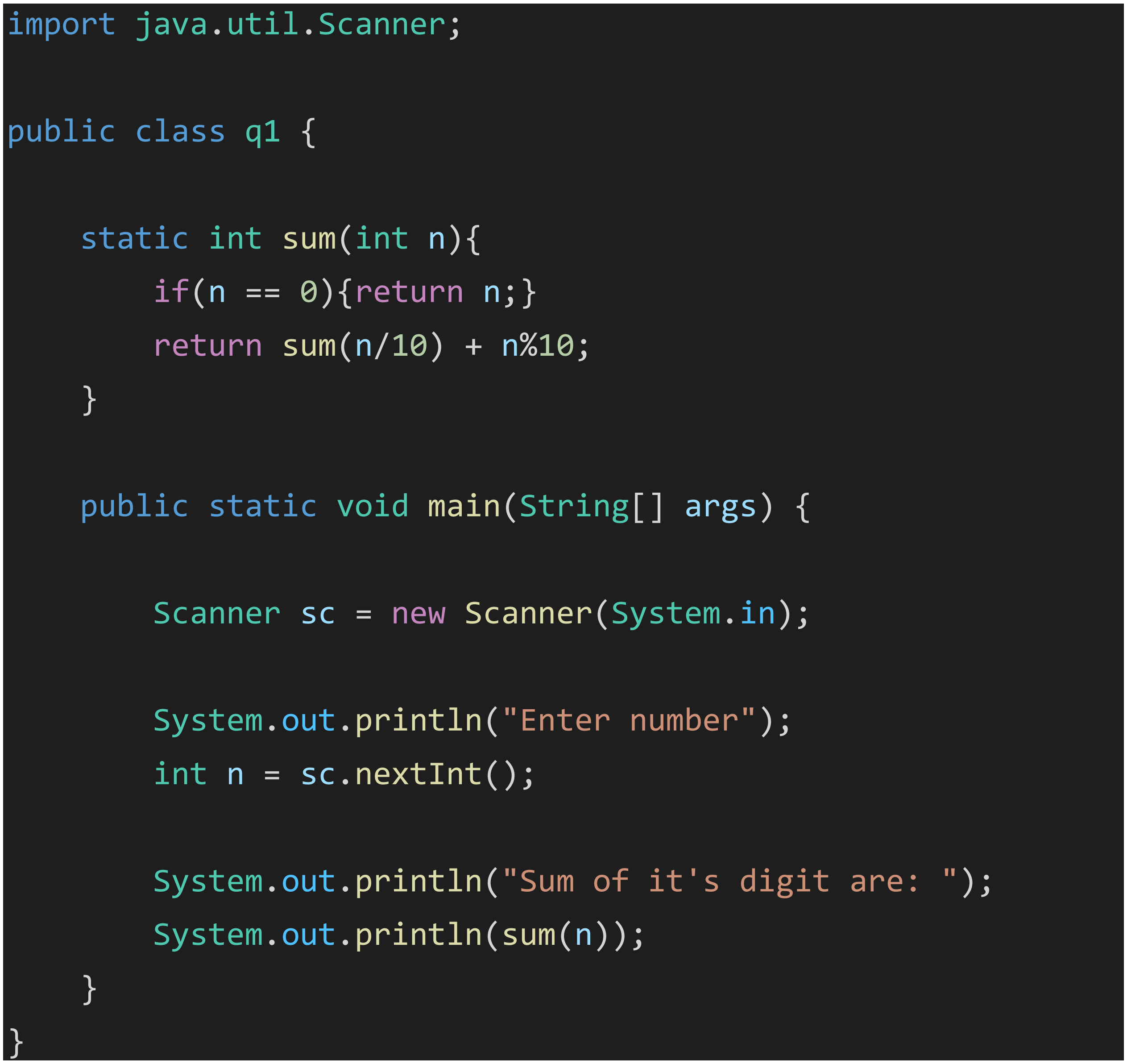
Recursion in java

Q1 : Given an integer, find out the sum of its digits using recursion.
Input: n= 1234
Output: 10
Explanation: 1+2+3+4=10



```java
import java.util.Scanner;

public class q1 {

    static int sum(int n){
        if(n == 0){return n;}
        return sum(n/10) + n%10;
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number");
        int n = sc.nextInt();

        System.out.println("Sum of it's digit are: ");
        System.out.println(sum(n));
    }
}
```

Q2: Given a number n. Find the sum of natural numbers till n but with alternate signs.

That means if n = 5 then you have to return 1-2+3-4+5 = 3 as your answer.

Constraints : 0<=n<=1e6

Input1 : n = 10

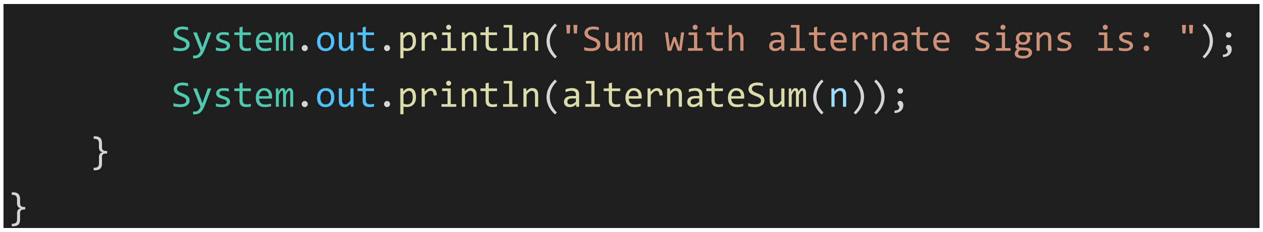
Output 1 : -5

Explanation : 1-2+3-4+5-6+7-8+9-10 = -5

Input 2 : n = 5

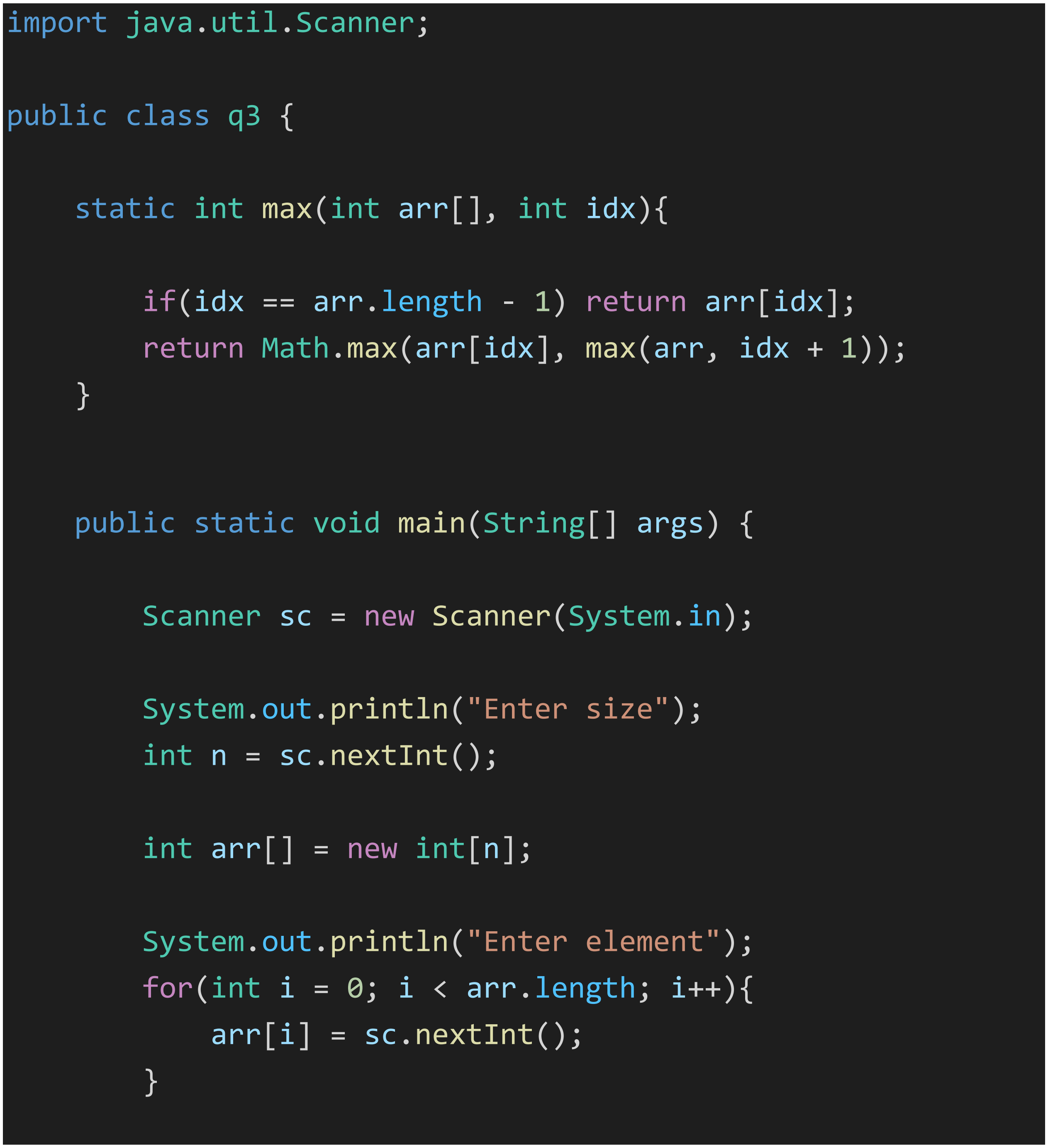
Output 2 : 3

```java
import java.util.Scanner;

public class q2 {

    static int alternateSum(int n){

        if(n == 1) return 1;
        if(n % 2 == 0){
            return alternateSum(n-1) - n;
        }
        else{
            return alternateSum(n-1) + n;
        }
    }


    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number");
        int n = sc.nextInt();
```



```java
        System.out.println("Sum with alternate signs is: ");
        System.out.println(alternateSum(n));
    }
}
```

Q3: Print the max value of the array [13, 1, -3, 22, 5].



```java
import java.util.Scanner;

public class q3 {

    static int max(int arr[], int idx){

        if(idx == arr.length - 1) return arr[idx];
        return Math.max(arr[idx], max(arr, idx + 1));
    }


    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter size");
        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter element");
        for(int i = 0; i < arr.length; i++){
            arr[i] = sc.nextInt();
        }

        System.out.println(max(arr,0));
    }
}
```

Q4 : Find the sum of the values of the array [92, 23, 15, -20, 10].

```
public class q4 {

    static int sumOfArray(int arr[], int idx){
        if(idx == arr.length-1) return arr[idx];
        return arr[idx] + sumOfArray(arr, idx + 1);

    }


    public static void main(String[] args) {

        int arr [] = {92, 23, 15, -20, 10};
        System.out.println(sumOfArray(arr,0));
    }
}
```

Q5. Given a number n. Print if it is an armstrong number or not. An armstrong number is a number if the sum of every digit in that number raised to the power of total digits in that number is equal to the number.

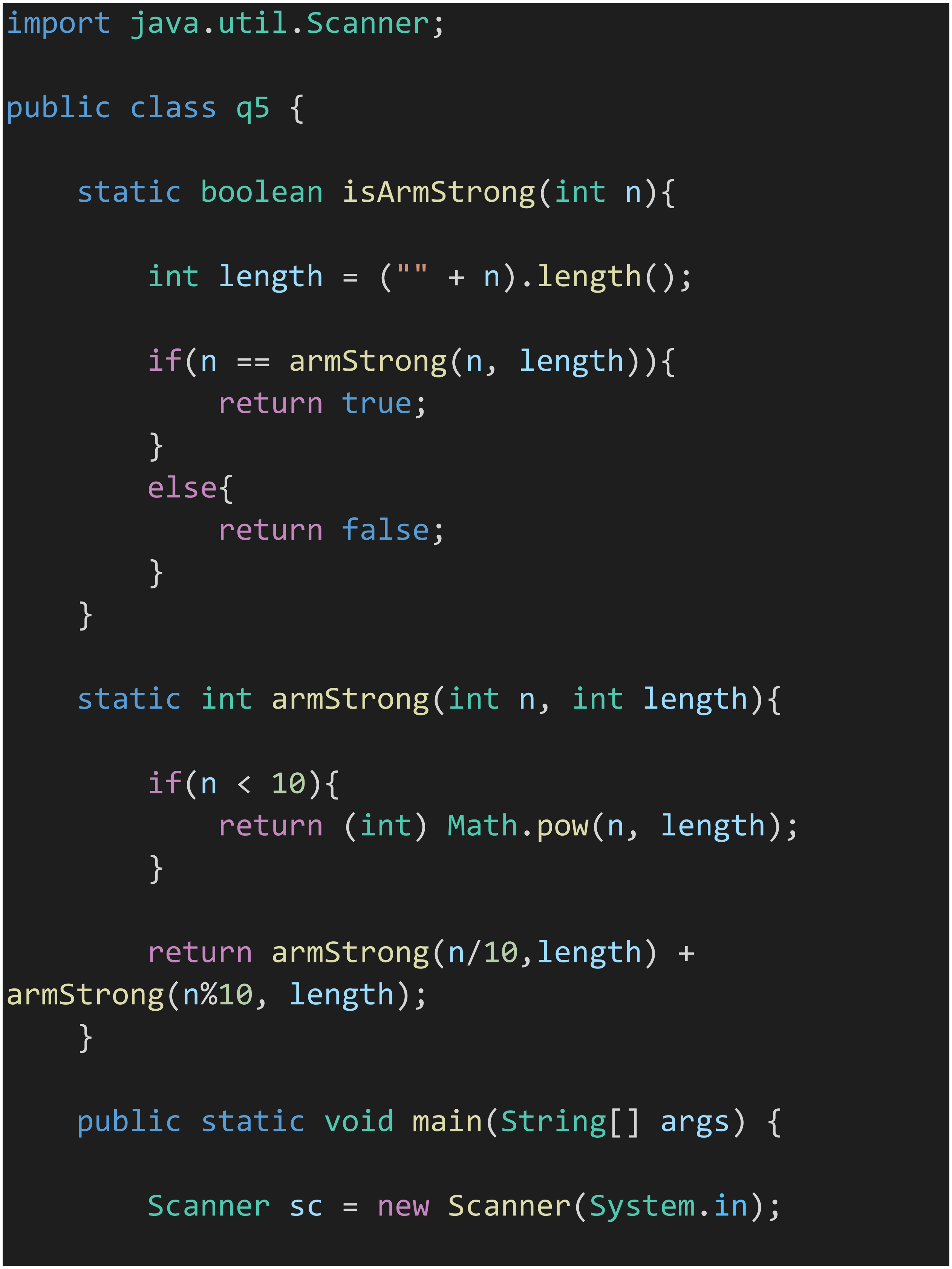
Example : 153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153 hence 153 is an armstrong number. (Easy)

Input1 : 153

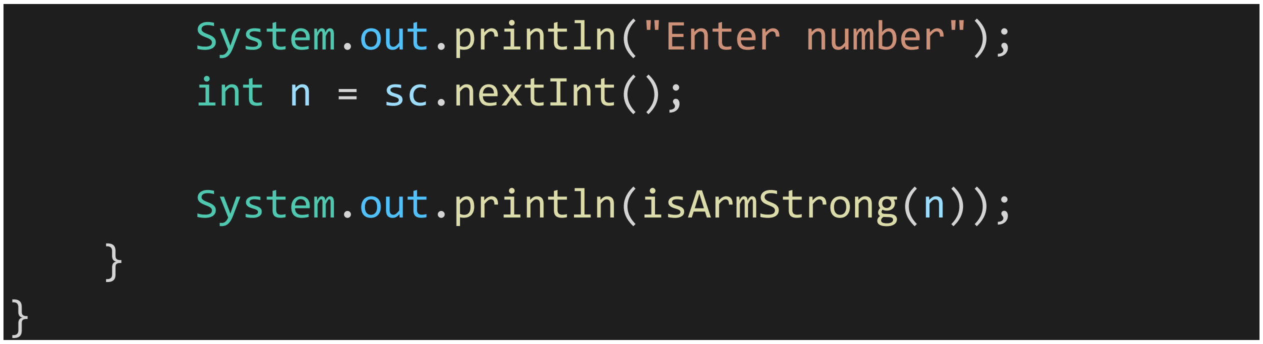
Output1 : Yes

Input 2 : 134

Output2 : No



```
import java.util.Scanner;

public class q5 {

    static boolean isArmStrong(int n){

        int length = ("" + n).length();

        if(n == armStrong(n, length)){
            return true;
        }
        else{
            return false;
        }
    }

    static int armStrong(int n, int length){

        if(n < 10){
            return (int) Math.pow(n, length);
        }

        return armStrong(n/10,length) +
armStrong(n%10, length);
    }

    public static void main(String[] args) {

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



```java
        System.out.println("Enter number");
        int n = sc.nextInt();

        System.out.println(isArmStrong(n));
    }
}
```