C-DAC Mumbai

Date 25/09/2024

Subject: Algorithm and Data Structure
Assignment 1

Name – Rohit Owal

Solve the assignment with following thing to be added in each question.

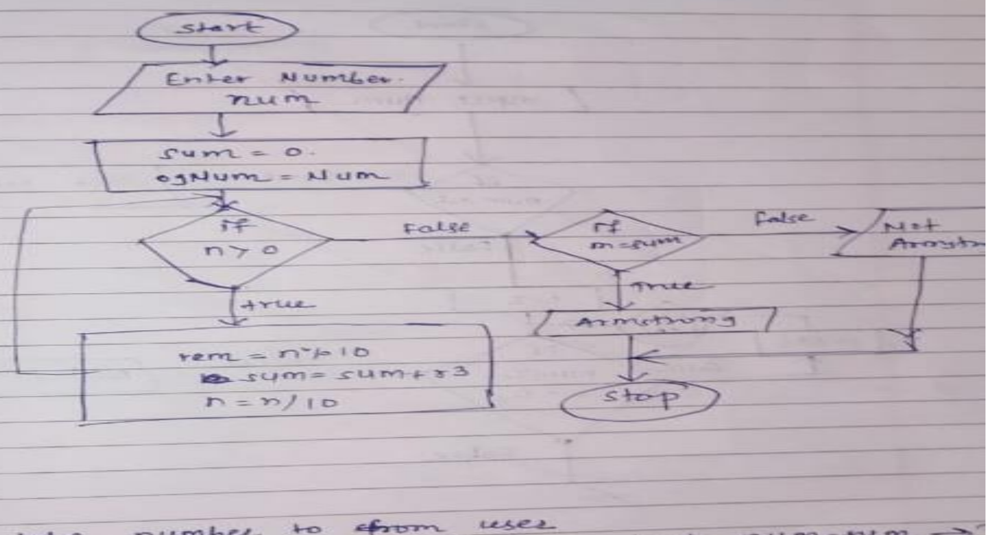
- -Program
- -Flow chart
- -Explanation
- -Output
- -Time and Space complexity

1. Armstrong Number
Problem: Write a Java program to check if a given number is an Armstrong number.

Test Cases:

Input: 153
Output: true
Input: 123
Output: false

1. Flowchart of Armstrong Number



Steps :-

1 → take number to ~~from~~ user

2 → check if (num > 0) $\xrightarrow{false}$ then check num = sum →

Arm

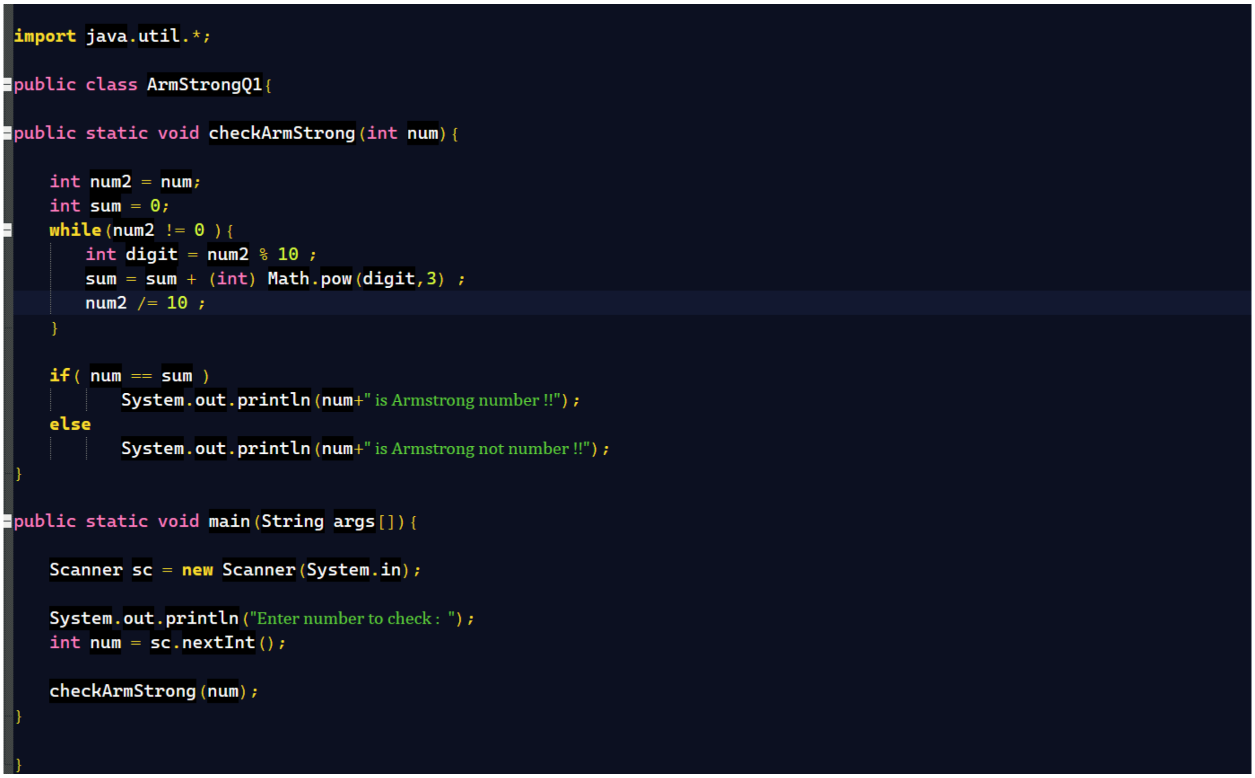
3 → If true then.

rem = n%10

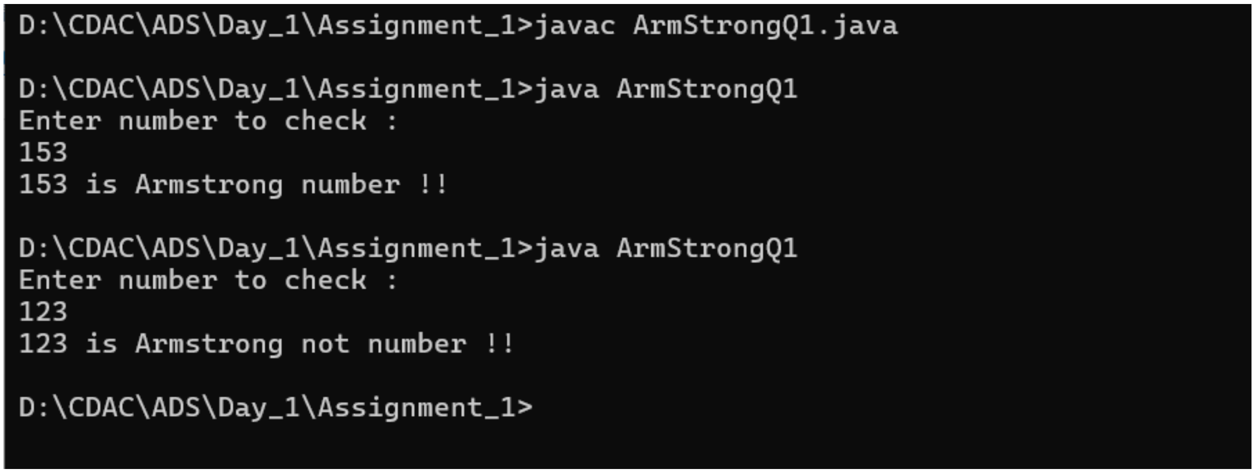
sum = sum + r3

n = n/10.

4 → go to step 2



```java
import java.util.*;

public class ArmStrongQ1{

public static void checkArmStrong(int num){

    int num2 = num;
    int sum = 0;
    while(num2 != 0 ){
        int digit = num2 % 10 ;
        sum = sum + (int) Math.pow(digit,3) ;
        num2 /= 10 ;
    }

    if( num == sum )
            System.out.println(num+" is Armstrong number !!");
    else
            System.out.println(num+" is Armstrong not number !!");
}

public static void main(String args[]){

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter number to check : ");
    int num = sc.nextInt();

    checkArmStrong(num);
}


}
```



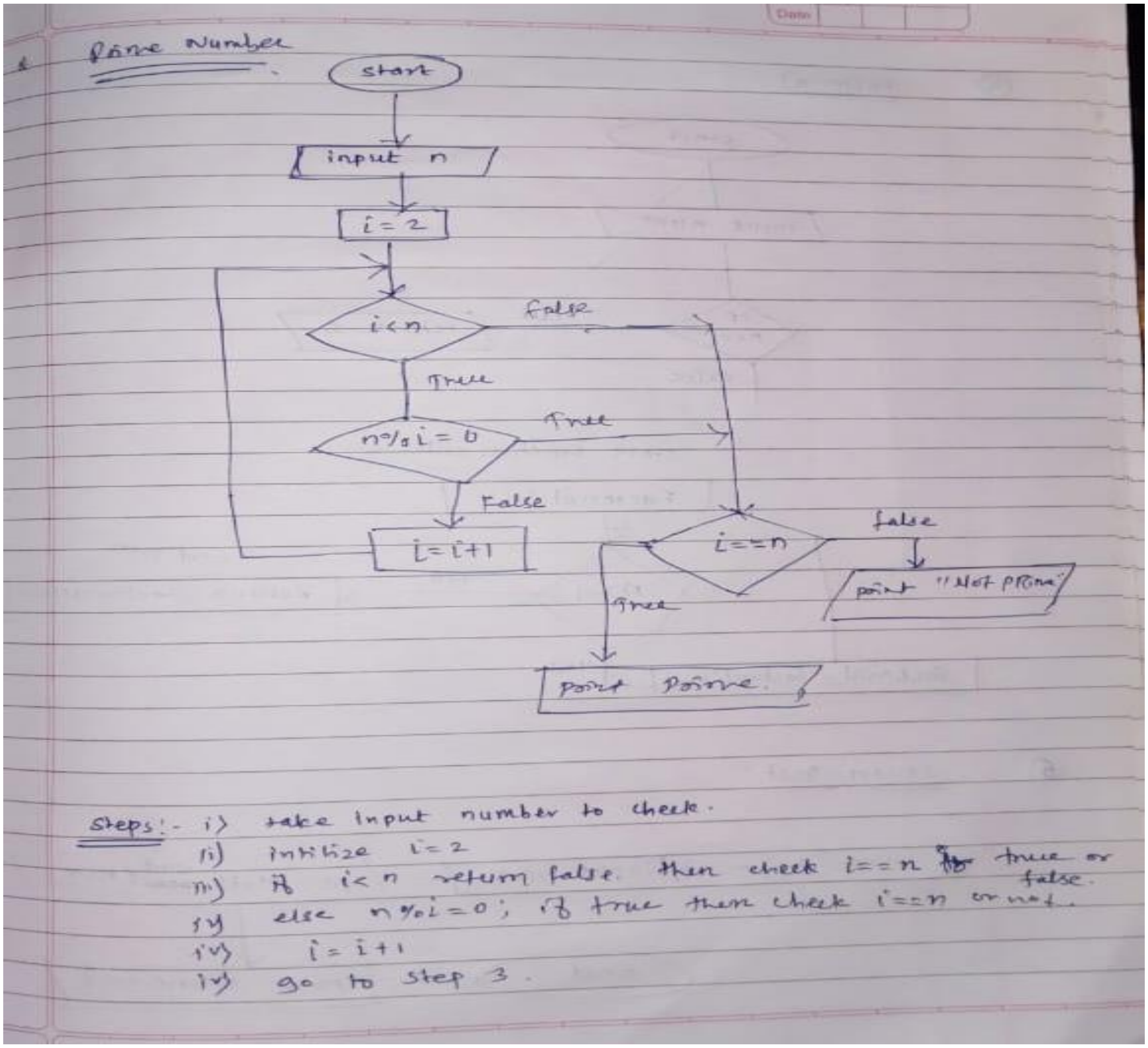
```
D:\CDAC\ADS\Day_1\Assignment_1>javac ArmStrongQ1.java

D:\CDAC\ADS\Day_1\Assignment_1>java ArmStrongQ1
Enter number to check :
153
153 is Armstrong number !!

D:\CDAC\ADS\Day_1\Assignment_1>java ArmStrongQ1
Enter number to check :
123
123 is Armstrong not number !!

D:\CDAC\ADS\Day_1\Assignment_1>
```

Time Complexity : O(logn)
Space Complexity: O(1)

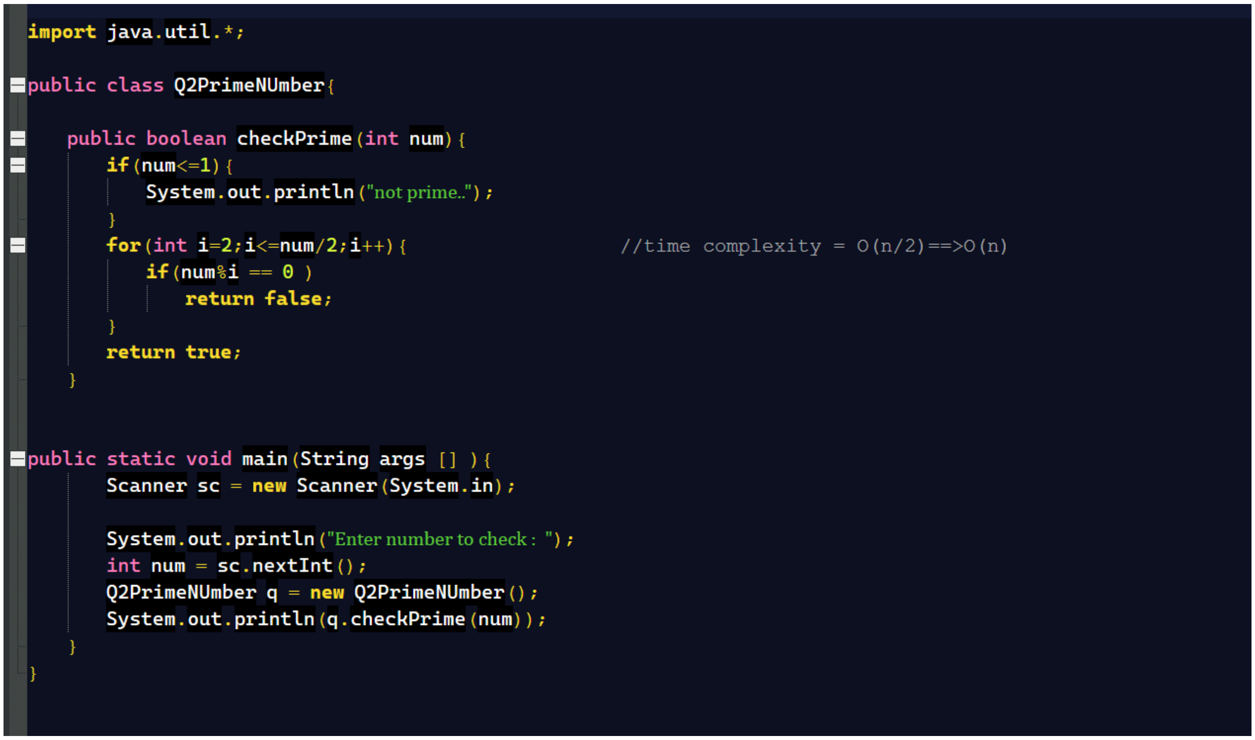
2. Prime Number
Problem: Write a Java program to check if a given number is prime.

Test Cases:

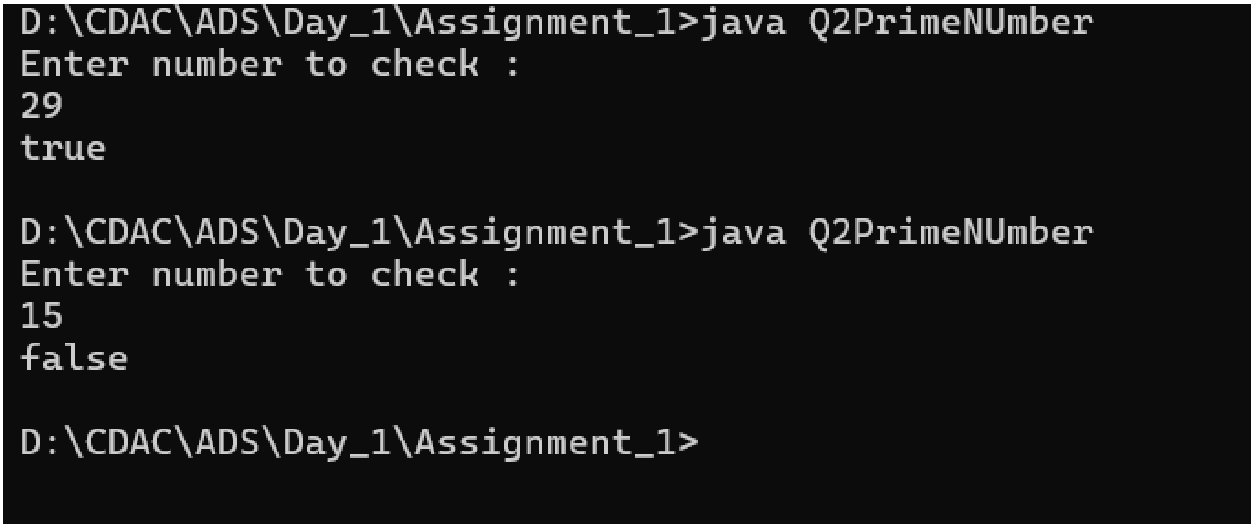
Input: 29
Output: true
Input: 15
Output: false



Steps:- i) take input number to check.
ii) initilize i = 2
iii) if i < n return false. then check i == n ~~fo~~ true or false.
iv) else n%i = 0; if true then check i==n or not.
iv) i = i+1
iv) go to step 3.



```java
import java.util.*;

public class Q2PrimeNUmber{

    public boolean checkPrime(int num){
        if(num<=1){
            System.out.println("not prime..");
        }
        for(int i=2;i<=num/2;i++){                    //time complexity = O(n/2)==>O(n)
            if(num%i == 0 )
                return false;
        }
        return true;
    }


public static void main(String args [] ){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number to check : ");
        int num = sc.nextInt();
        Q2PrimeNUmber q = new Q2PrimeNUmber();
        System.out.println(q.checkPrime(num));
    }
}
```



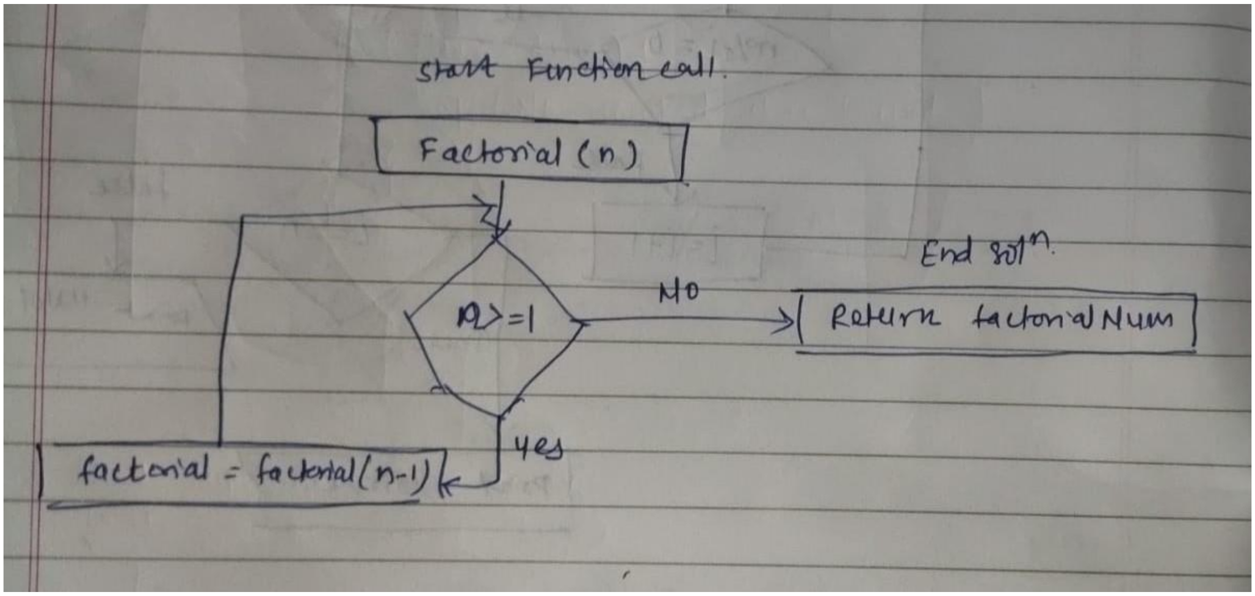
```
D:\CDAC\ADS\Day_1\Assignment_1>java Q2PrimeNUmber
Enter number to check :
29
true

D:\CDAC\ADS\Day_1\Assignment_1>java Q2PrimeNUmber
Enter number to check :
15
false

D:\CDAC\ADS\Day_1\Assignment_1>
```

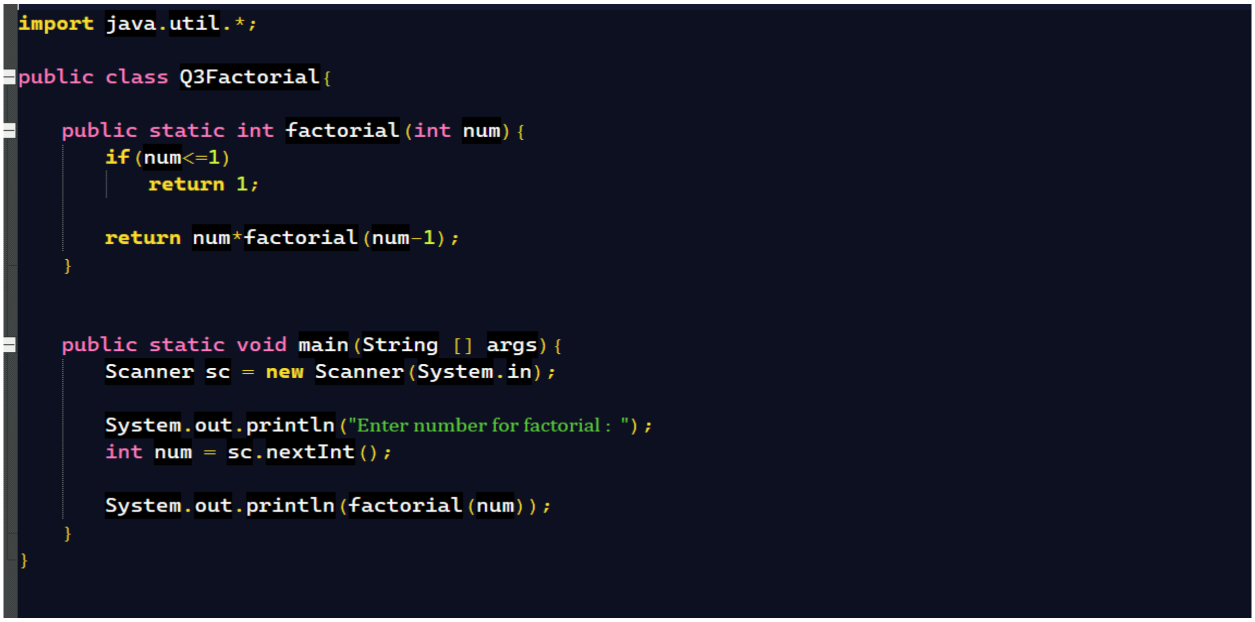
Time Complexity : O(n)
Space Complexity: O(1)

3. Factorial
Problem: Write a Java program to compute the factorial of a given number.

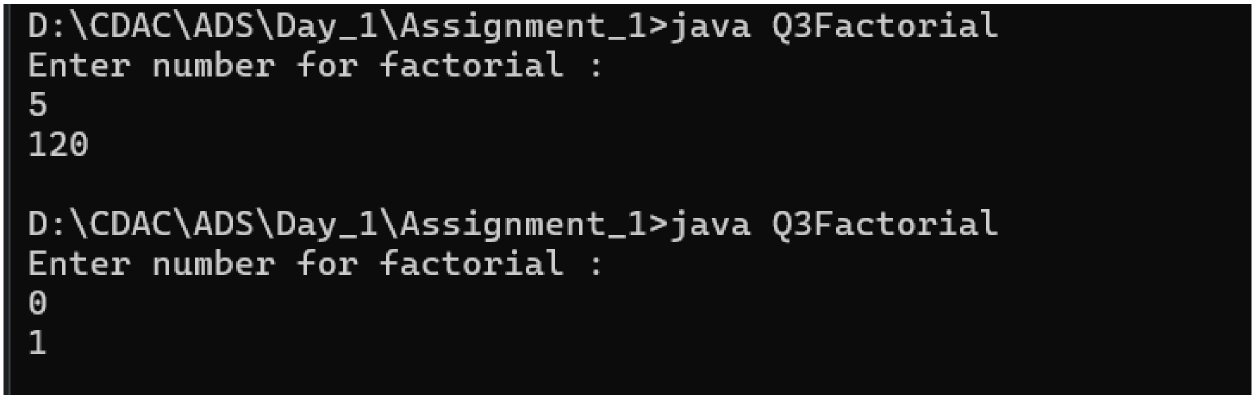
Test Cases:

Input: 5
Output: 120
Input: 0
Output: 1





```
import java.util.*;

public class Q3Factorial{

    public static int factorial(int num){
        if(num<=1)
            return 1;

        return num*factorial(num-1);
    }


    public static void main(String [] args){
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number for factorial : ");
        int num = sc.nextInt();

        System.out.println(factorial(num));
    }
}
```



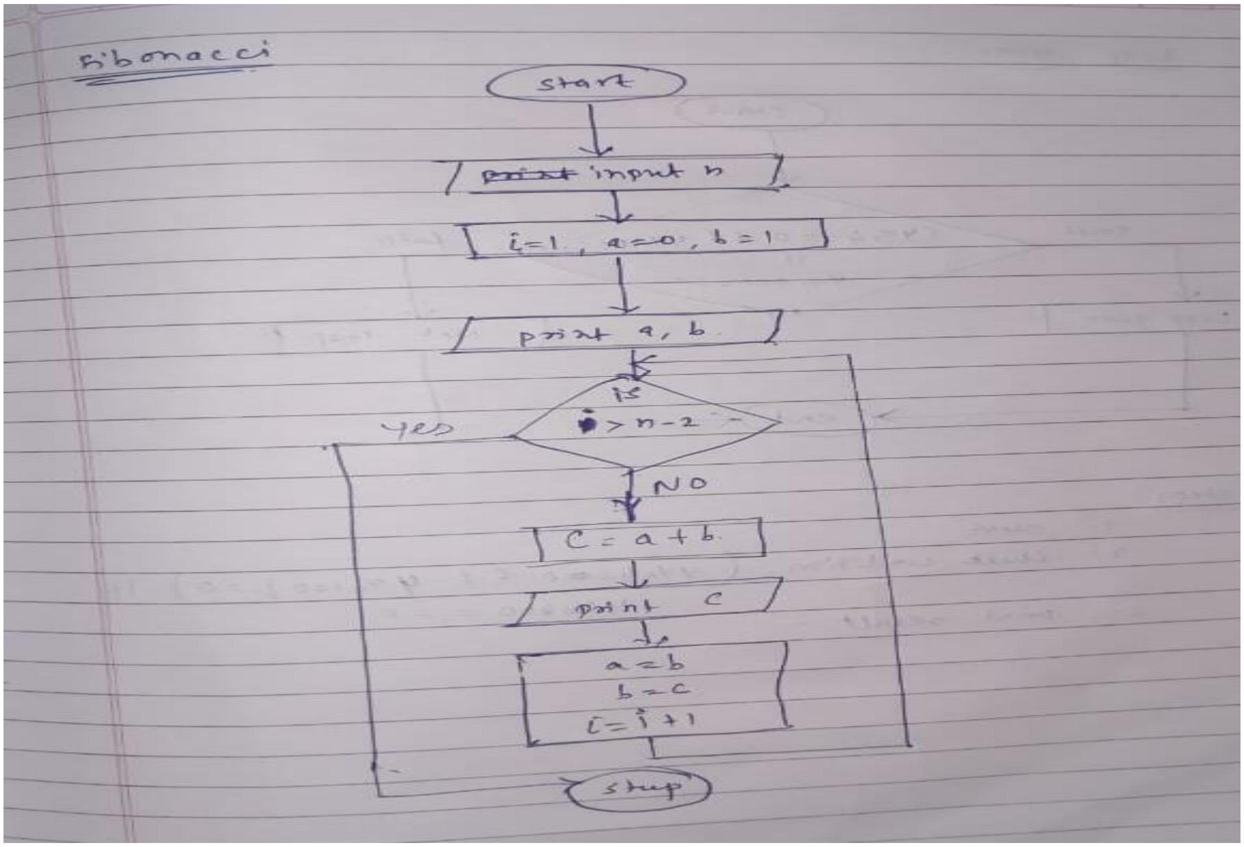
```
D:\CDAC\ADS\Day_1\Assignment_1>java Q3Factorial
Enter number for factorial :
5
120

D:\CDAC\ADS\Day_1\Assignment_1>java Q3Factorial
Enter number for factorial :
0
1
```

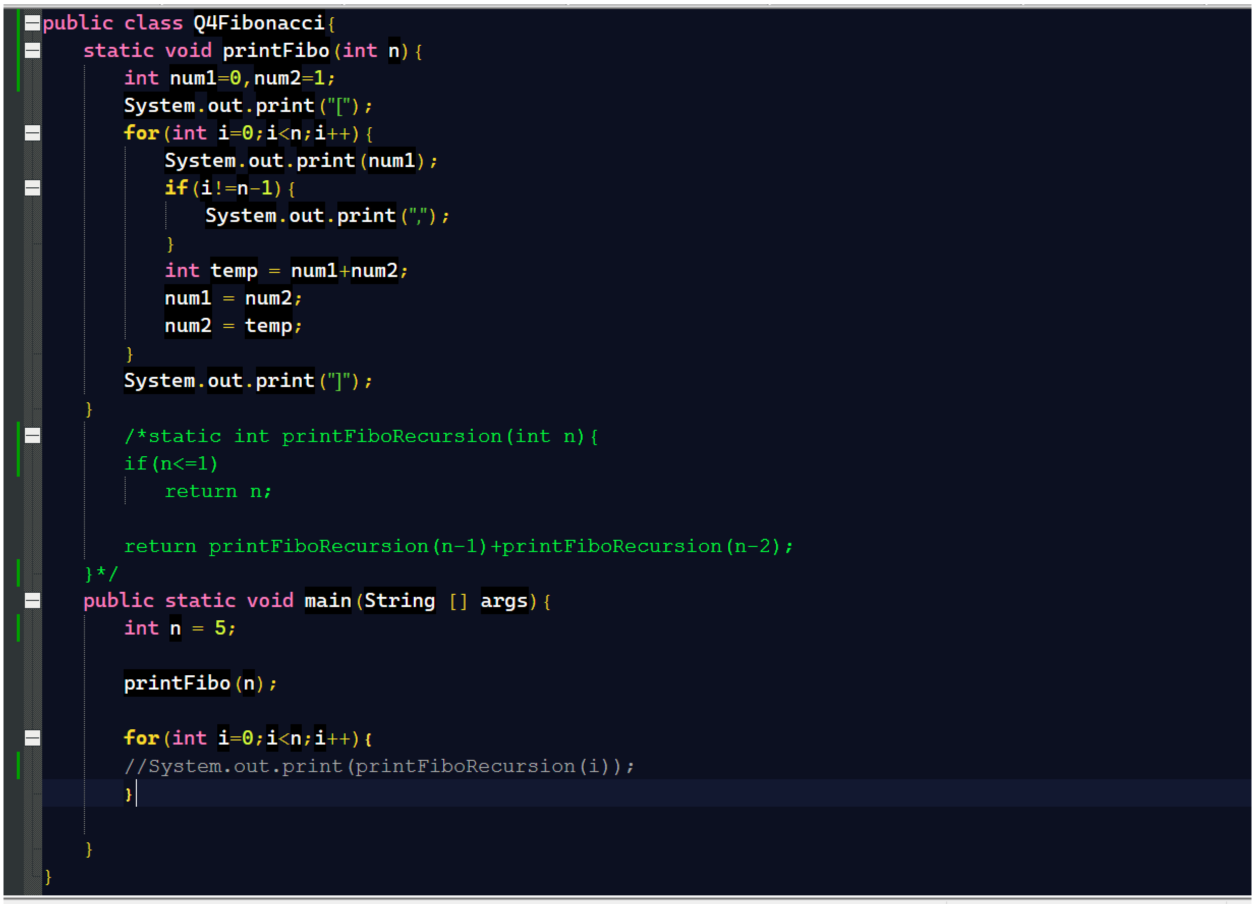
Time Complexity : O(n)
Space Complexity: O(n)

4. Fibonacci Series
Problem: Write a Java program to print the first n numbers in the Fibonacci series.

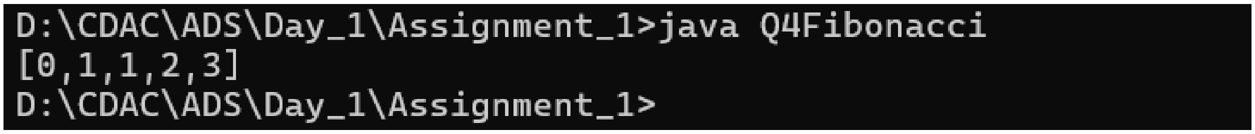
Test Cases:

Input: n = 5
Output: [0, 1, 1, 2, 3]
Input: n = 8
Output: [0, 1, 1, 2, 3, 5, 8, 13]





```java
public class Q4Fibonacci{
    static void printFibo(int n){
        int num1=0,num2=1;
        System.out.print("[");
        for(int i=0;i<n;i++){
            System.out.print(num1);
            if(i!=n-1){
                System.out.print(",");
            }
            int temp = num1+num2;
            num1 = num2;
            num2 = temp;
        }
        System.out.print("]");
    }
        /*static int printFiboRecursion(int n){
        if(n<=1)
            return n;

        return printFiboRecursion(n-1)+printFiboRecursion(n-2);
    }*/
    public static void main(String [] args){
        int n = 5;

        printFibo(n);

        for(int i=0;i<n;i++){
        //System.out.print(printFiboRecursion(i));
        }

    }
}
```



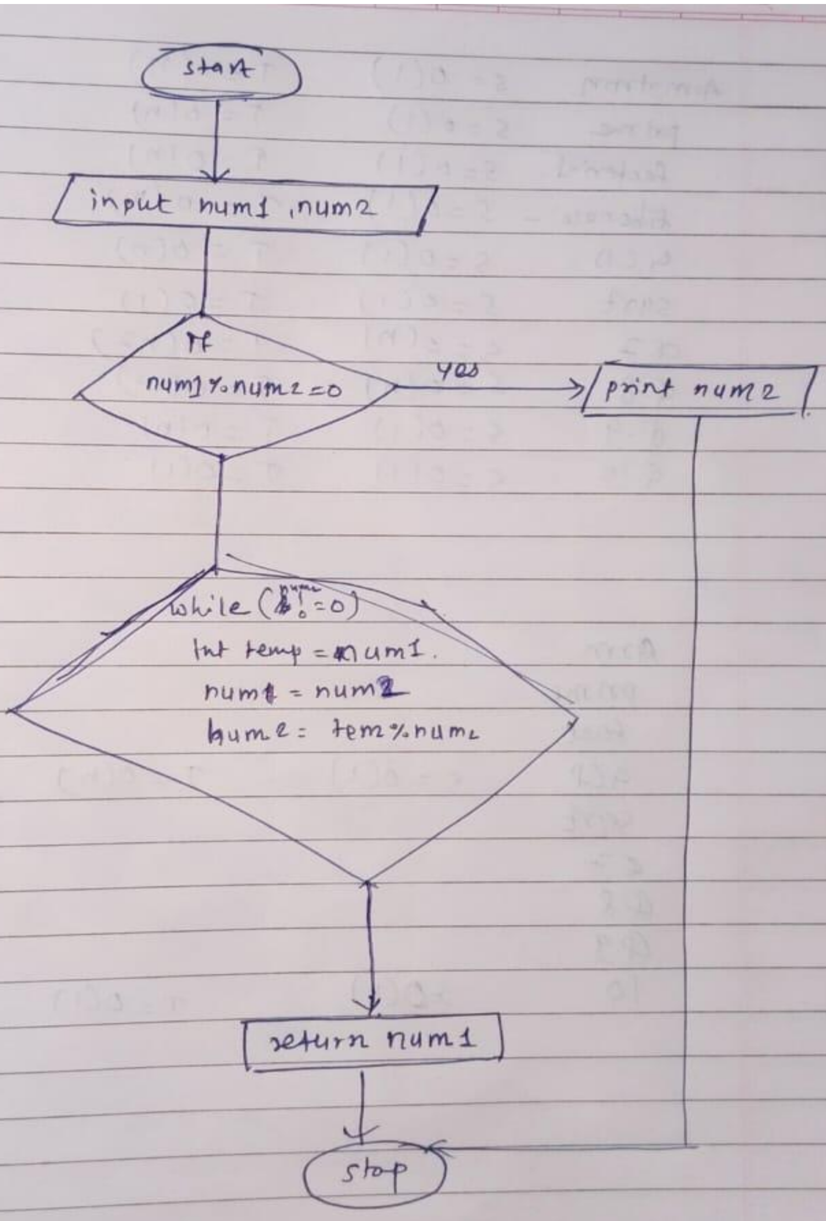
```
D:\CDAC\ADS\Day_1\Assignment_1>java Q4Fibonacci
[0,1,1,2,3]
D:\CDAC\ADS\Day_1\Assignment_1>
```

Time Complexity : O(n)
Space Complexity: O(1)

5. Find GCD
Problem: Write a Java program to find the Greatest Common Divisor (GCD) of two numbers.

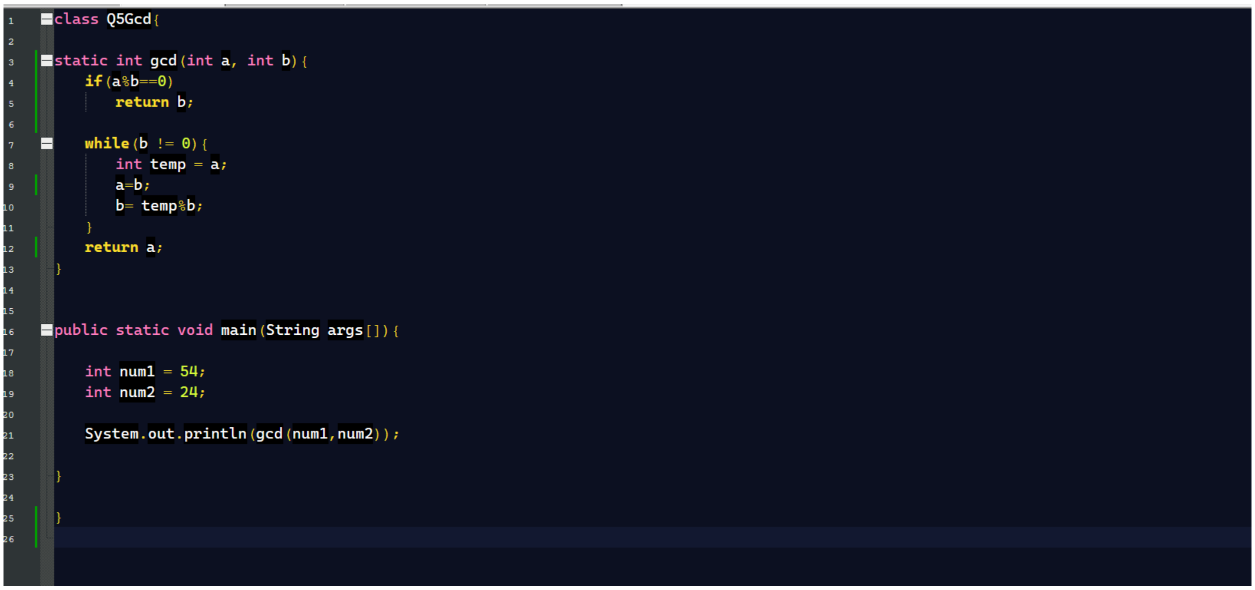
Test Cases:
Input: a = 54, b = 24
Output: 6
Input: a = 17, b = 13
Output: 1

Q.5 GCD →

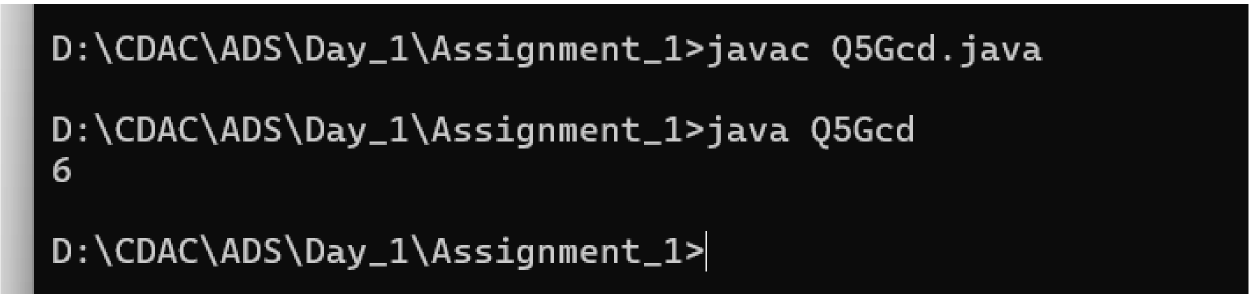


Steps →

```
1. input num1 & num2;
2. if num1 % num2 → return num 2
3. else.   while ( nu2 != 0)
                int temp = num1
                num1 = num2
                num2 = temp % num2
4. return num1.
```



```
class Q5Gcd{

static int gcd(int a, int b){
    if(a%b==0)
        return b;

    while(b != 0){
        int temp = a;
        a=b;
        b= temp%b;
    }
    return a;
}


public static void main(String args[]){

    int num1 = 54;
    int num2 = 24;

    System.out.println(gcd(num1,num2));

}

}
```



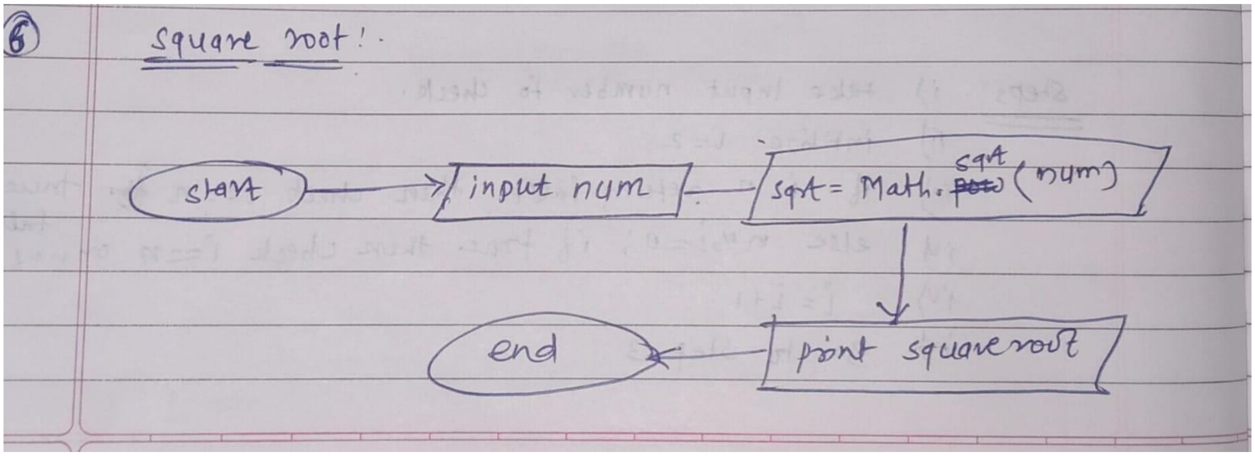
```
D:\CDAC\ADS\Day_1\Assignment_1>javac Q5Gcd.java

D:\CDAC\ADS\Day_1\Assignment_1>java Q5Gcd
6

D:\CDAC\ADS\Day_1\Assignment_1>
```

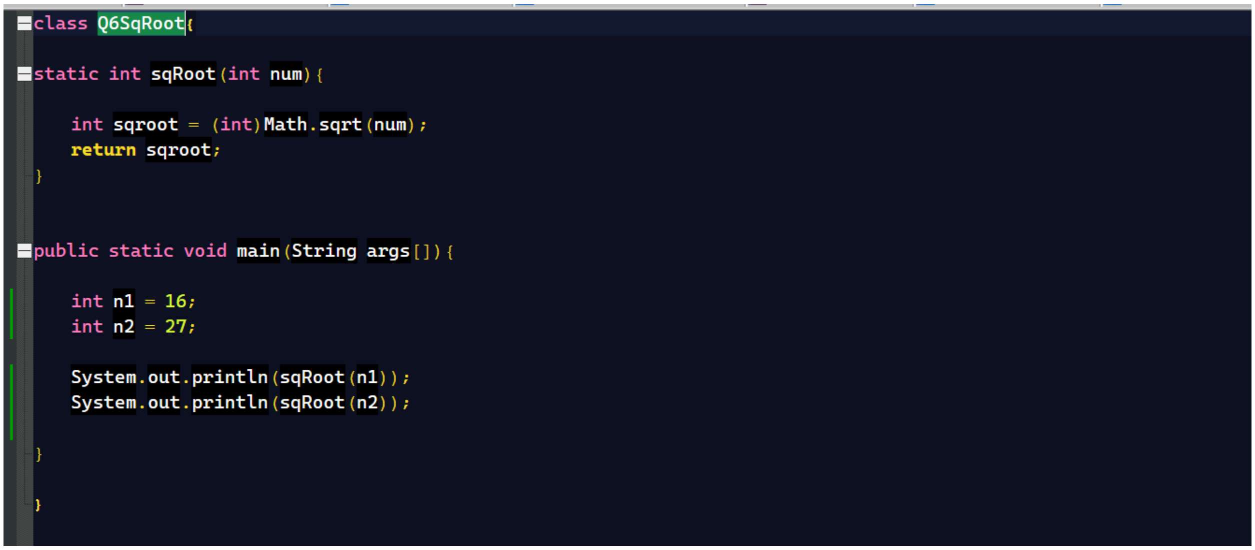
Time Complexity: O(log(min(a,b)))
Space Complexity: O(1)

6. Find Square Root
Problem: Write a Java program to find the square root of a given number (using integer approximation).

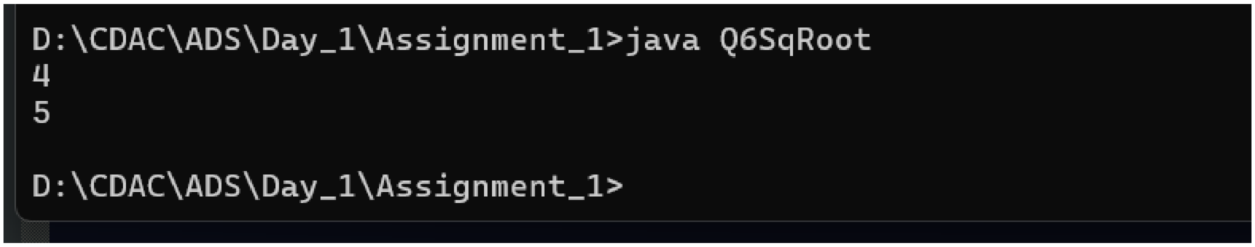
Test Cases:

Input: x = 16
Output: 4
Input: x = 27
Output: 5





```
class Q6SqRoot{

static int sqRoot(int num){

    int sqroot = (int)Math.sqrt(num);
    return sqroot;
}


public static void main(String args[]){

    int n1 = 16;
    int n2 = 27;

    System.out.println(sqRoot(n1));
    System.out.println(sqRoot(n2));

}

}
```



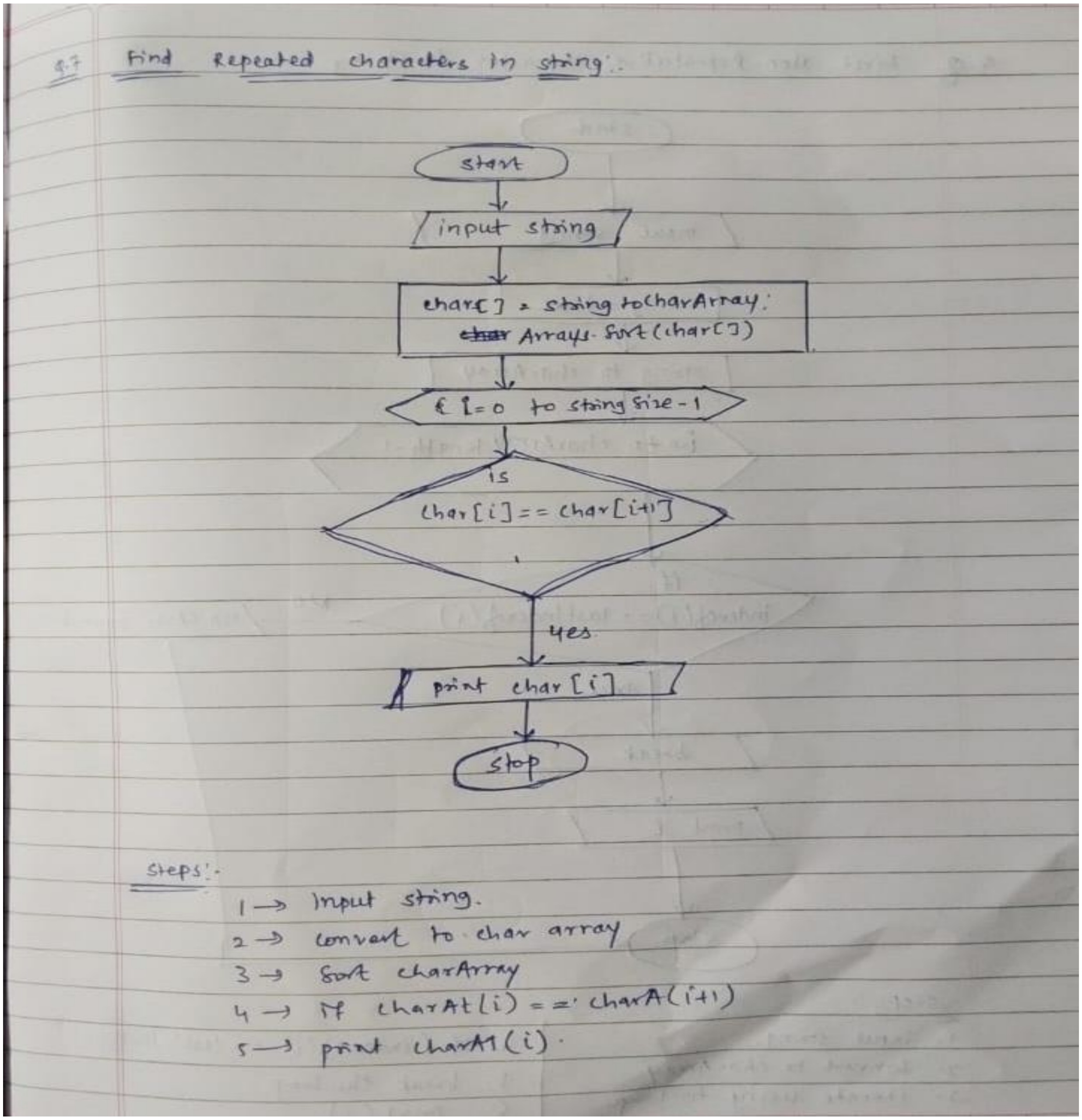
```
D:\CDAC\ADS\Day_1\Assignment_1>java Q6SqRoot
4
5

D:\CDAC\ADS\Day_1\Assignment_1>
```

Time Complexity: O(1)
Space Complexity: O(1)

7. Find Repeated Characters in a String
Problem: Write a Java program to find all repeated characters in a string.

Test Cases:

Input: "programming"
Output: ['r', 'g', 'm']
Input: "hello"
Output: ['l']

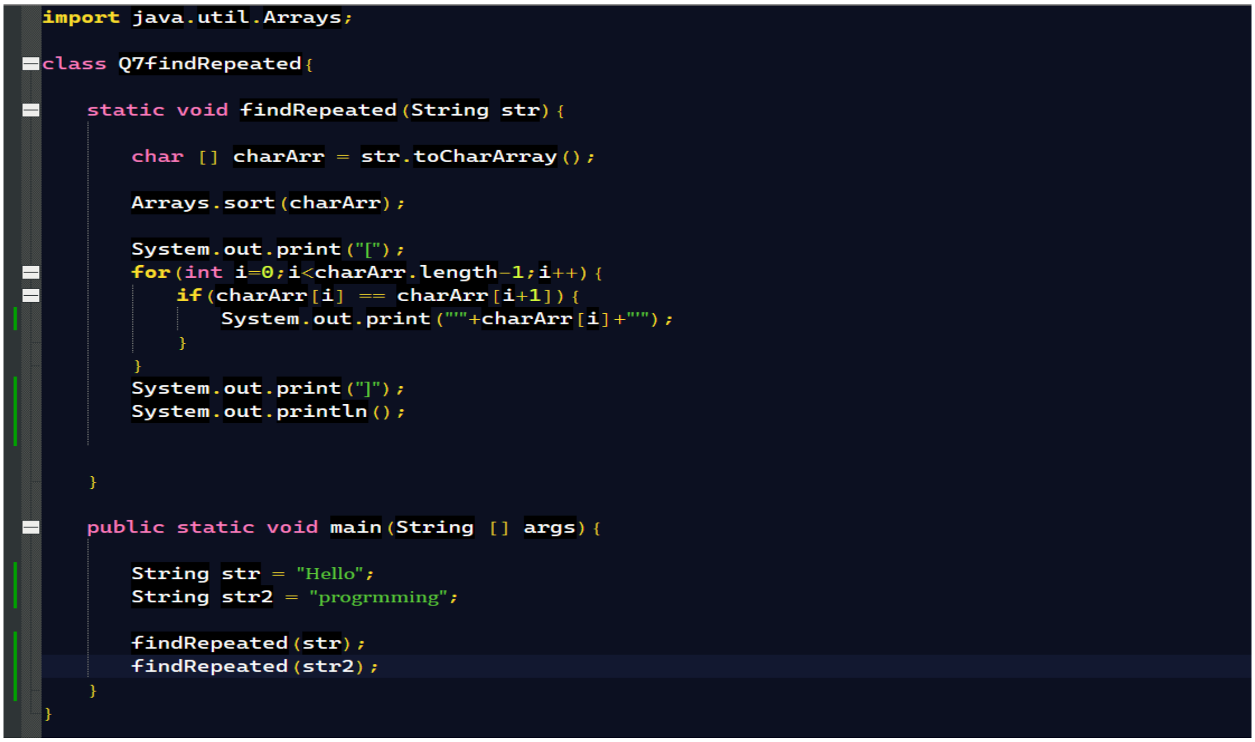


Q.7 Find Repeated characters in string:-

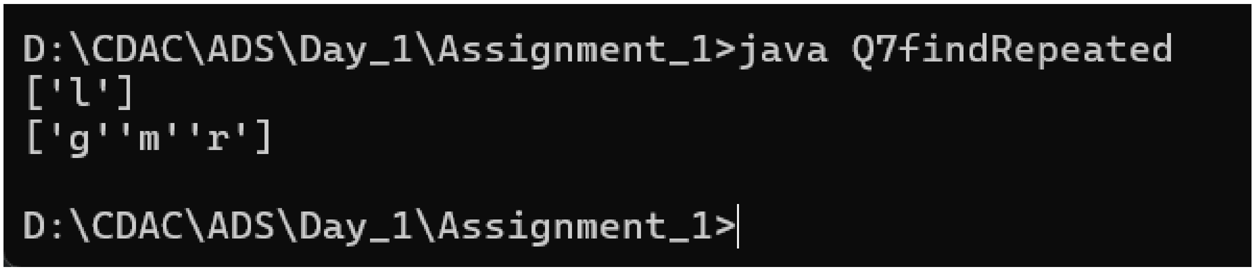
start
input string
char[] = string toCharArray;
~~char~~ Arrays. Sort (char[])
{ i = 0 to string size - 1
is
char[i] == char[i+1]
yes
print char [i]
stop

Steps:-

1 → Input string.
2 → convert to char array
3 → Sort charArray
4 → If charAt(i) ==' charA(i+1)
5 → print charAt(i).



```java
import java.util.Arrays;

class Q7findRepeated{

    static void findRepeated(String str){

        char [] charArr = str.toCharArray();

        Arrays.sort(charArr);

        System.out.print("[");
        for(int i=0;i<charArr.length-1;i++){
            if(charArr[i] == charArr[i+1]){
                System.out.print("'"+charArr[i]+"'");
            }
        }
        System.out.print("]");
        System.out.println();


    }

    public static void main(String [] args){

        String str = "Hello";
        String str2 = "progrmming";

        findRepeated(str);
        findRepeated(str2);
    }
}
```



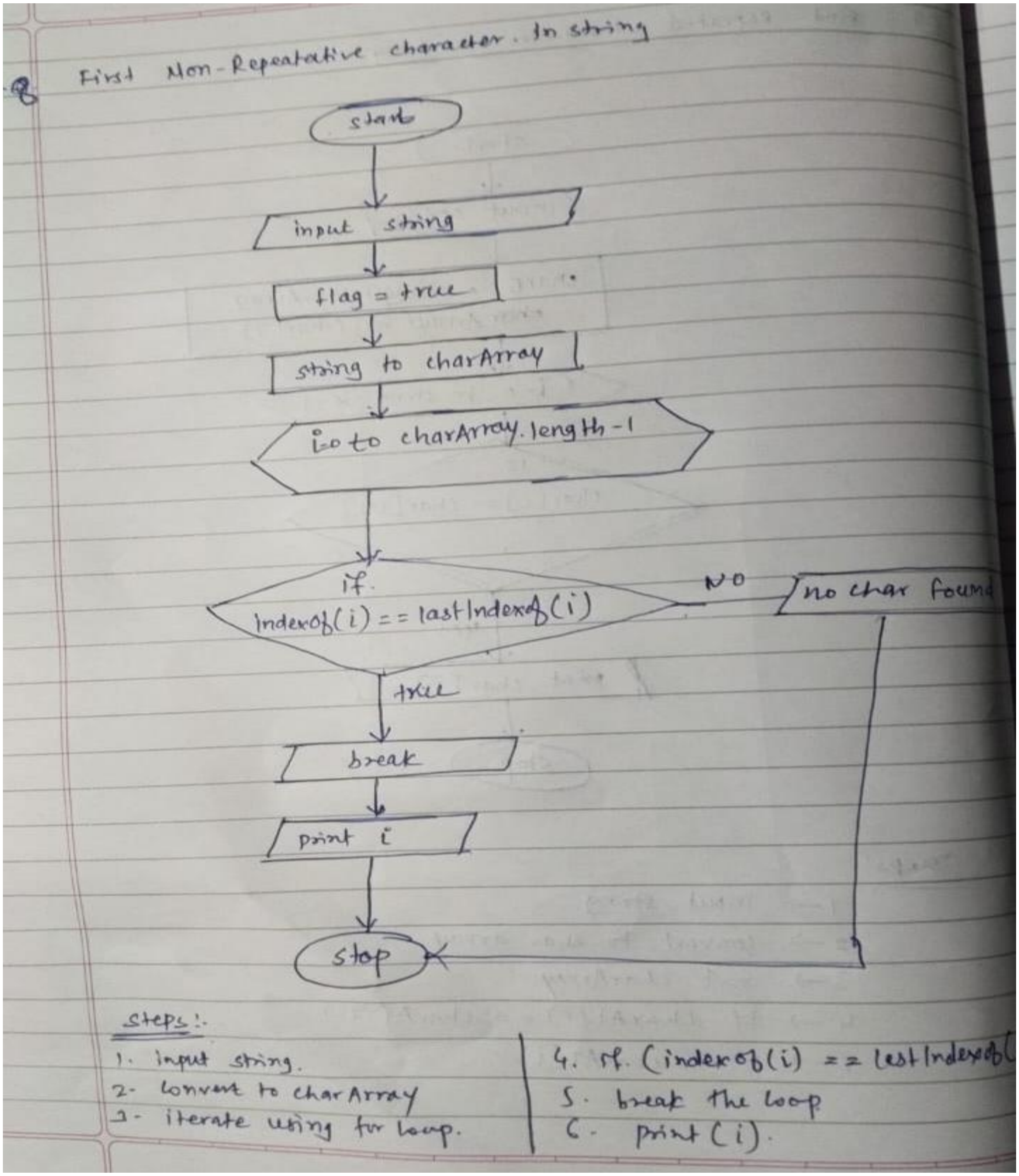
```
D:\CDAC\ADS\Day_1\Assignment_1>java Q7findRepeated
['l']
['g''m''r']

D:\CDAC\ADS\Day_1\Assignment_1>
```

Time Complexity: O(nlogn)
Space Complexity: O(n)

8. First Non-Repeated Character
Problem: Write a Java program to find the first non-repeated character in a string.

Test Cases:

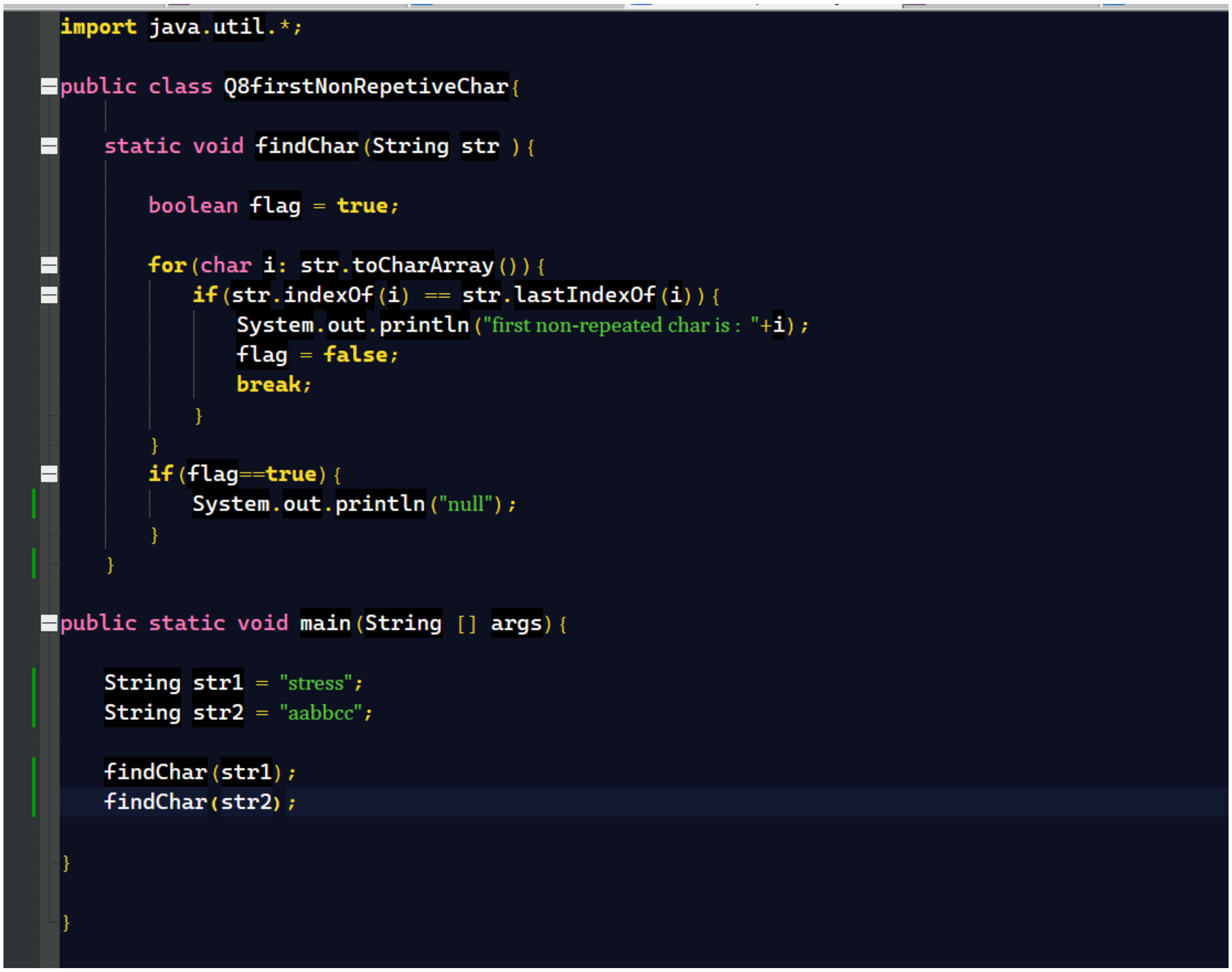
Input: "stress"
Output: 't'
Input: "aabbcc"
Output: null



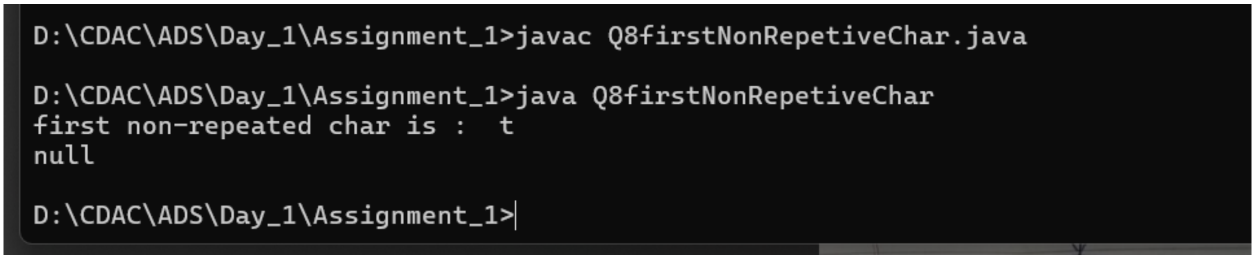
Q First Non-Repeatative character. In string

Steps:-

1. input string.
2. convert to charArray
3. iterate using for loop.
4. if. (indexof(i) == lastIndexof(
5. break the loop
6. print (i).



```
import java.util.*;

public class Q8firstNonRepetiveChar{

    static void findChar(String str ){

        boolean flag = true;

        for(char i: str.toCharArray()){
            if(str.indexOf(i) == str.lastIndexOf(i)){
                System.out.println("first non-repeated char is : "+i);
                flag = false;
                break;
            }
        }
        if(flag==true){
            System.out.println("null");
        }
    }

public static void main(String [] args){

    String str1 = "stress";
    String str2 = "aabbcc";

    findChar(str1);
    findChar(str2);

}

}
```



```
D:\CDAC\ADS\Day_1\Assignment_1>javac Q8firstNonRepetiveChar.java

D:\CDAC\ADS\Day_1\Assignment_1>java Q8firstNonRepetiveChar
first non-repeated char is :  t
null

D:\CDAC\ADS\Day_1\Assignment_1>
```

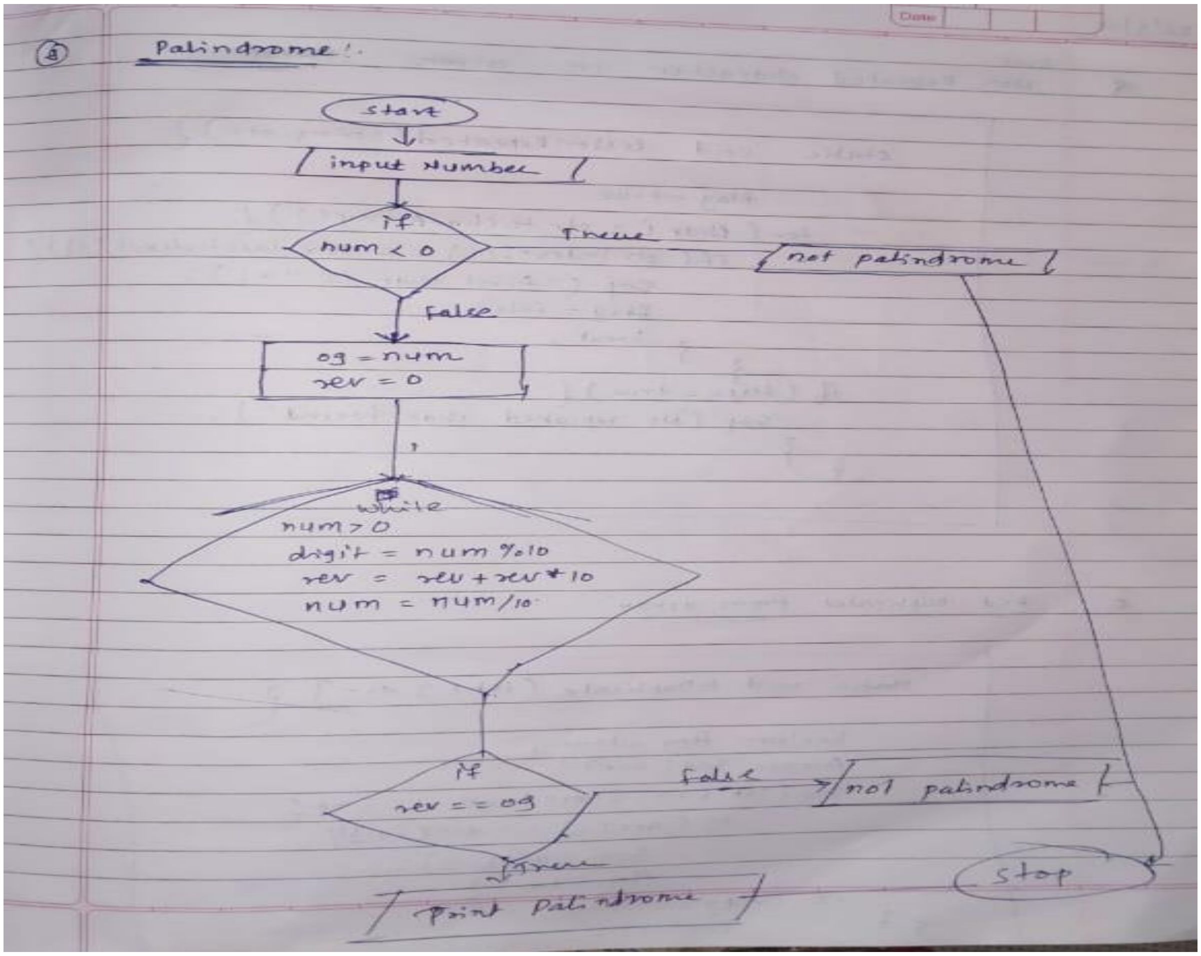
Time Complexity: O(n^2)
Space Complexity: O(n)

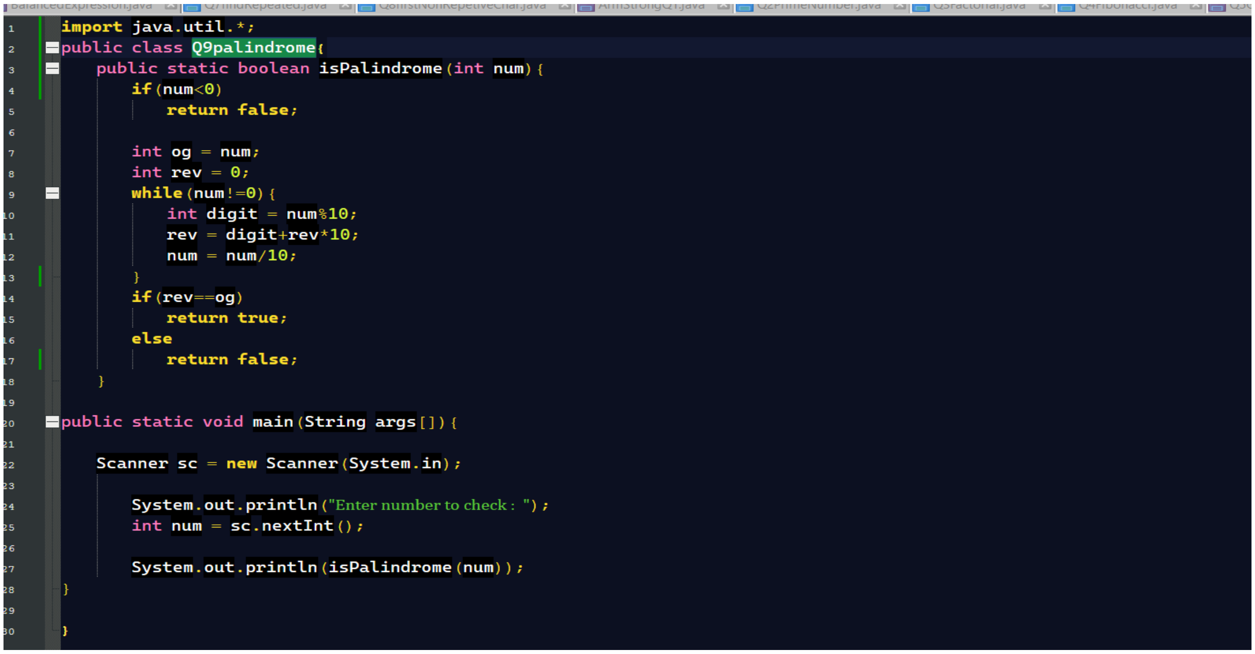
9. Integer Palindrome
Problem: Write a Java program to check if a given integer is a palindrome.

Test Cases:

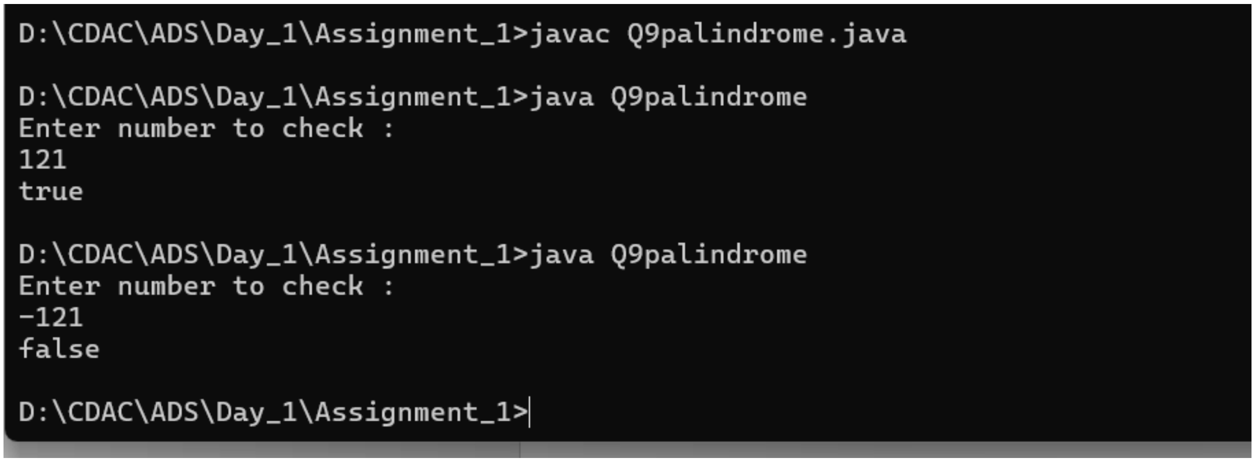
Input: 121

Output: true
Input: -121
Output: false





```java
import java.util.*;
public class Q9palindrome{
    public static boolean isPalindrome(int num){
        if(num<0)
            return false;

        int og = num;
        int rev = 0;
        while(num!=0){
            int digit = num%10;
            rev = digit+rev*10;
            num = num/10;
        }
        if(rev==og)
            return true;
        else
            return false;
    }

public static void main(String args[]){

    Scanner sc = new Scanner(System.in);

        System.out.println("Enter number to check : ");
        int num = sc.nextInt();

        System.out.println(isPalindrome(num));
}

}
```



```
D:\CDAC\ADS\Day_1\Assignment_1>javac Q9palindrome.java

D:\CDAC\ADS\Day_1\Assignment_1>java Q9palindrome
Enter number to check :
121
true

D:\CDAC\ADS\Day_1\Assignment_1>java Q9palindrome
Enter number to check :
-121
false

D:\CDAC\ADS\Day_1\Assignment_1>
```

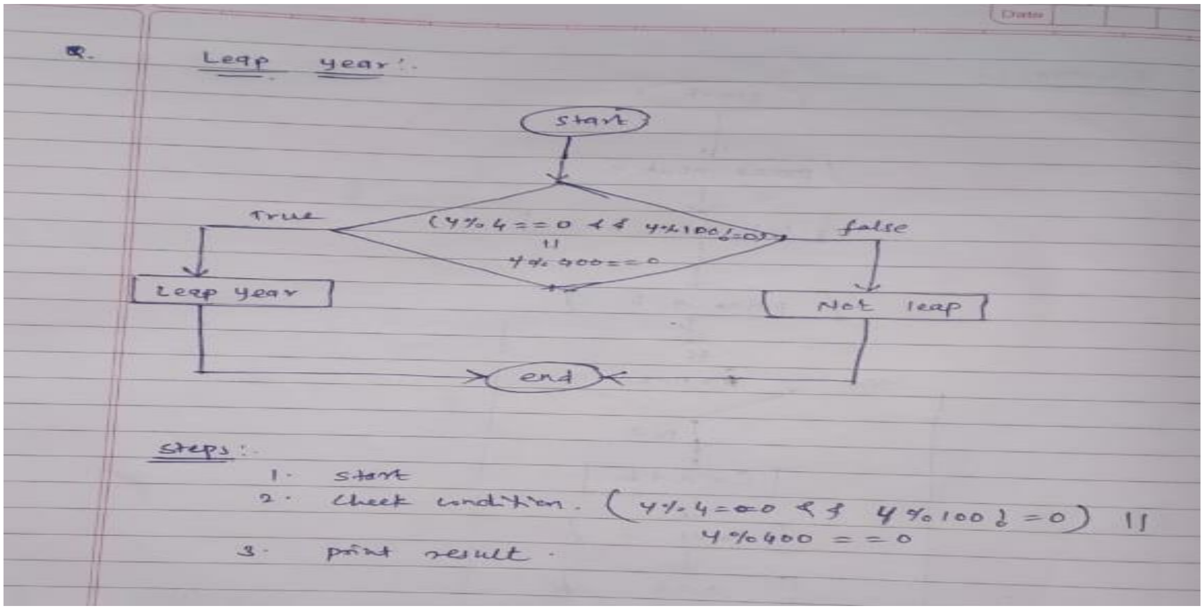
Time Complexity : O(log n)
Space Complexity: O(1)

10. Leap Year

Problem: Write a Java program to check if a given year is a leap year.

Test Cases:

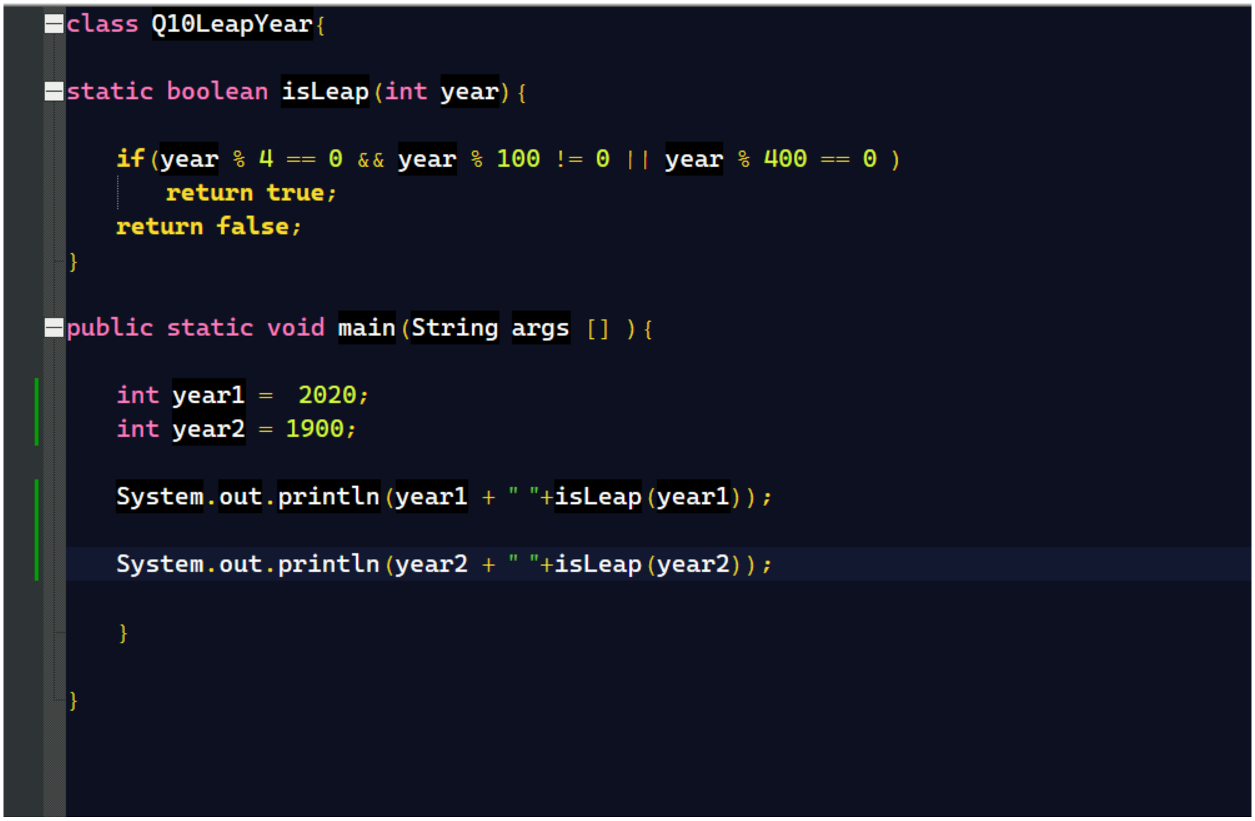
Input: 2020
Output: true
Input: 1900
Output: false



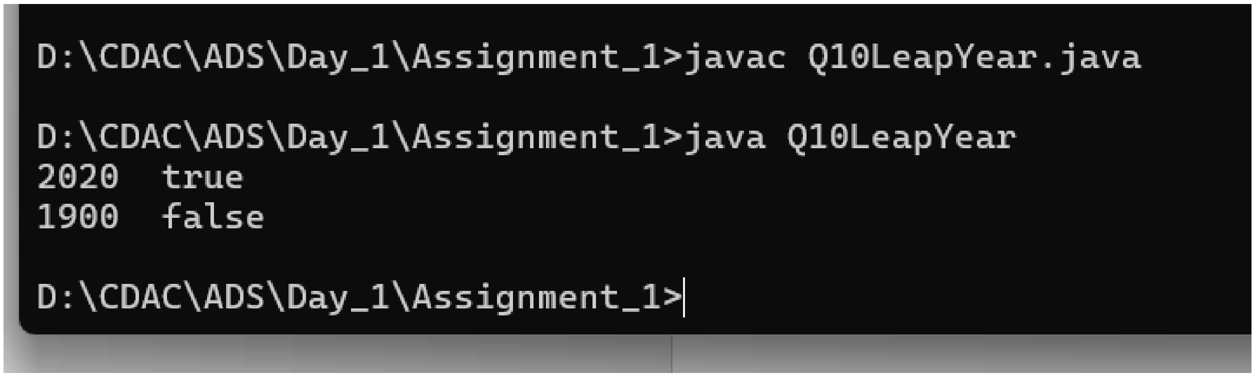
Q. Leap year :-

Steps :-

1. start
2. Check condition. (y%4==0 && y%100!=0) ||
   y%400==0
3. print result.



```java
class Q10LeapYear{

static boolean isLeap(int year){

    if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0 )
        return true;
    return false;
}

public static void main(String args [] ){

    int year1 =  2020;
    int year2 = 1900;

    System.out.println(year1 + " "+isLeap(year1));

    System.out.println(year2 + " "+isLeap(year2));

    }

}
```



```
D:\CDAC\ADS\Day_1\Assignment_1>javac Q10LeapYear.java

D:\CDAC\ADS\Day_1\Assignment_1>java Q10LeapYear
2020  true
1900  false

D:\CDAC\ADS\Day_1\Assignment_1>
```

Time Complexity : O(1)
Space Complexity: O(1)