Date: 13th April, 2021

-Programming Fundamentals

-Lab Tasks

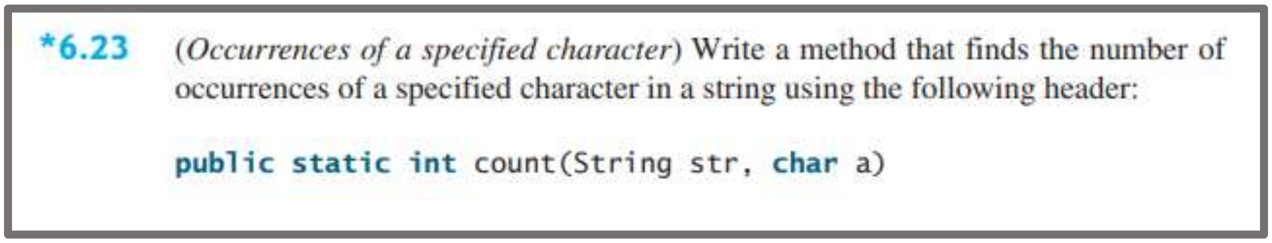
-Submitted to: Sir Azfar Shakeel Khan

-Roll # FA20-BSE-094

-M.Ruslan Babar.

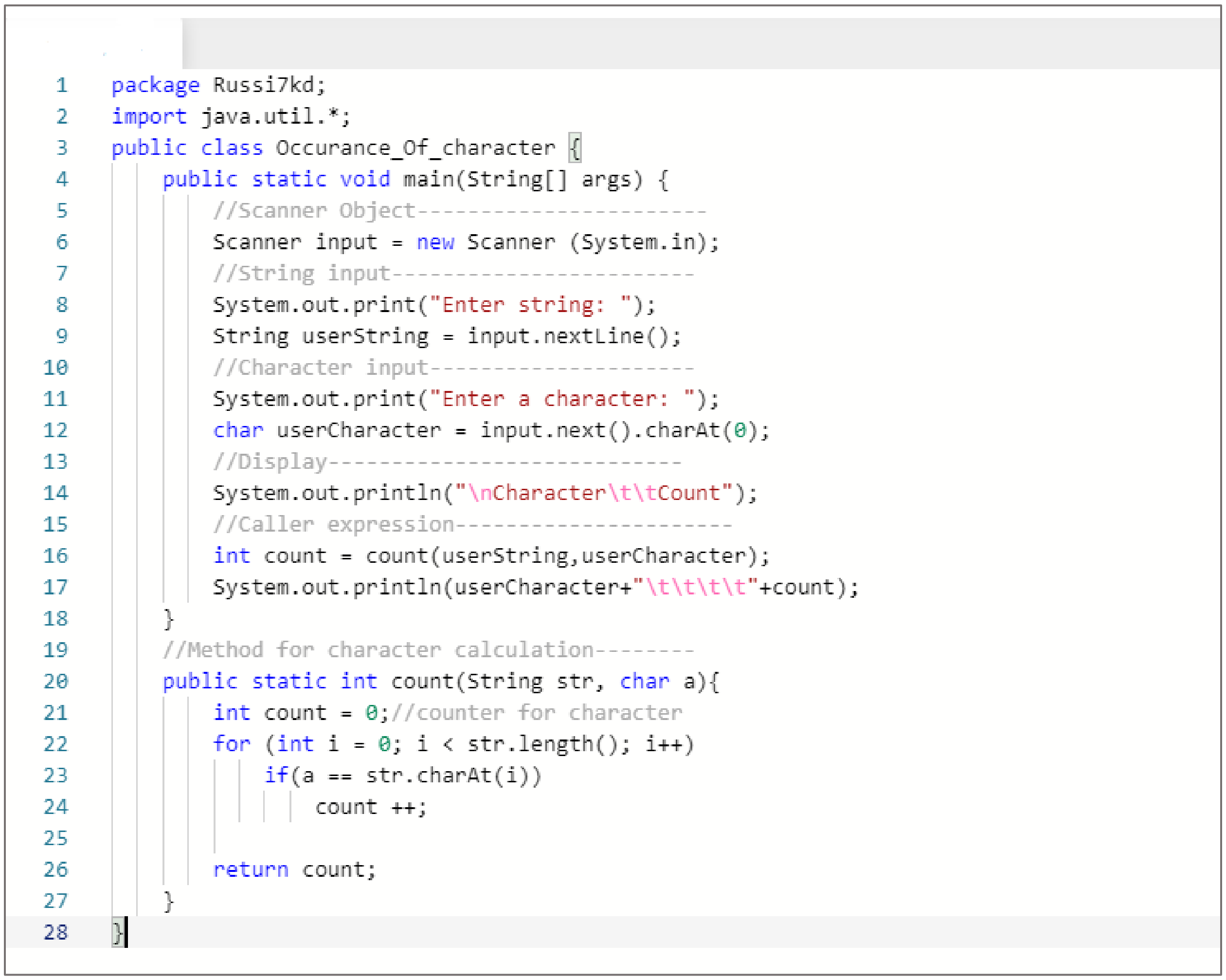
-Date: 13/04/2021

Question 6.23



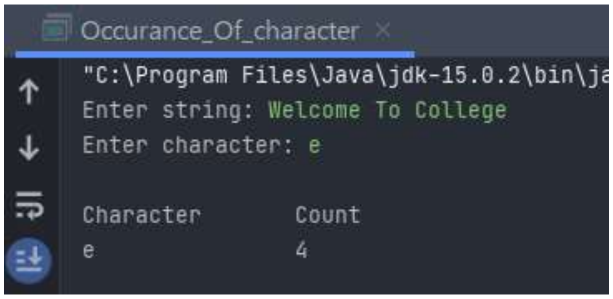
***6.23** (*Occurrences of a specified character*) Write a method that finds the number of occurrences of a specified character in a string using the following header:

```java
public static int count(String str, char a)
```



```java
package Russi7kd;
import java.util.*;
public class Occurance_Of_character {
    public static void main(String[] args) {
        //Scanner Object----------------------
        Scanner input = new Scanner (System.in);
        //String input-----------------------
        System.out.print("Enter string: ");
        String userString = input.nextLine();
        //Character input--------------------
        System.out.print("Enter a character: ");
        char userCharacter = input.next().charAt(0);
        //Display--------------------------
        System.out.println("\nCharacter\t\tCount");
        //Caller expression---------------------
        int count = count(userString,userCharacter);
        System.out.println(userCharacter+"\t\t\t\t"+count);
    }
    //Method for character calculation-------
    public static int count(String str, char a){
        int count = 0;//counter for character
        for (int i = 0; i < str.length(); i++)
            if(a == str.charAt(i))
                count ++;

        return count;
    }
}
```

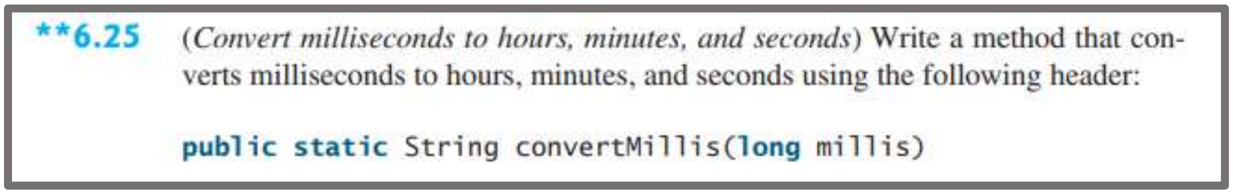
OUTPUT 6.23:



Occurance_Of_character

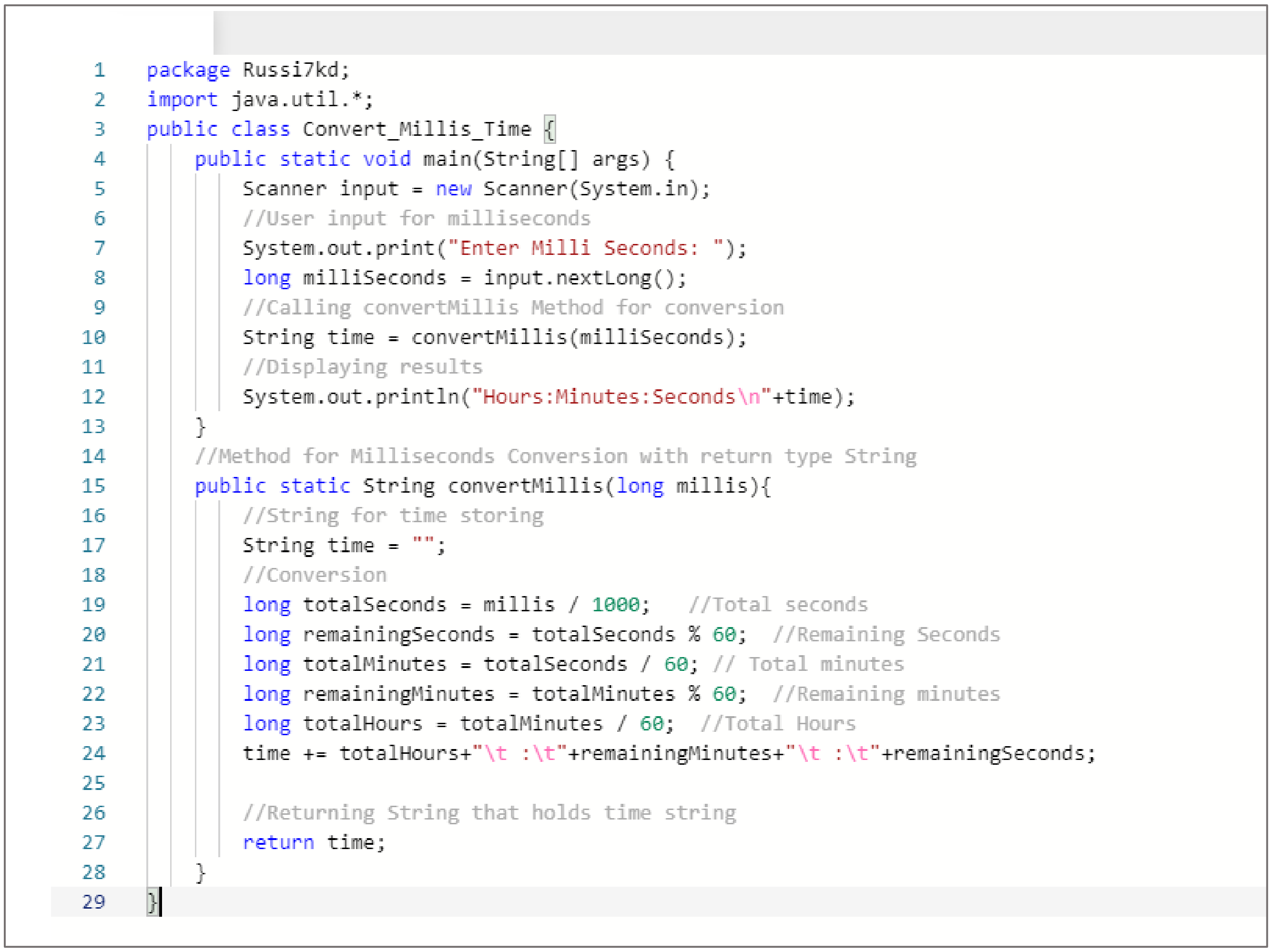
```
"C:\Program Files\Java\jdk-15.0.2\bin\ja
Enter string: Welcome To College
Enter character: e

Character       Count
e               4
```

Question 6.25



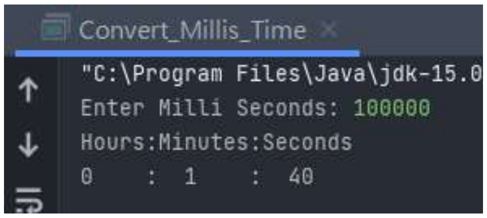
****6.25** (*Convert milliseconds to hours, minutes, and seconds*) Write a method that converts milliseconds to hours, minutes, and seconds using the following header:

```java
public static String convertMillis(long millis)
```

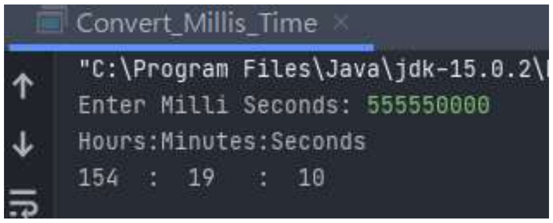


```java
package Russi7kd;
import java.util.*;
public class Convert_Millis_Time {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        //User input for milliseconds
        System.out.print("Enter Milli Seconds: ");
        long milliSeconds = input.nextLong();
        //Calling convertMillis Method for conversion
        String time = convertMillis(milliSeconds);
        //Displaying results
        System.out.println("Hours:Minutes:Seconds\n"+time);
    }
    //Method for Milliseconds Conversion with return type String
    public static String convertMillis(long millis){
        //String for time storing
        String time = "";
        //Conversion
        long totalSeconds = millis / 1000;   //Total seconds
        long remainingSeconds = totalSeconds % 60;  //Remaining Seconds
        long totalMinutes = totalSeconds / 60; // Total minutes
        long remainingMinutes = totalMinutes % 60;  //Remaining minutes
        long totalHours = totalMinutes / 60;  //Total Hours
        time += totalHours+"\t :\t"+remainingMinutes+"\t :\t"+remainingSeconds;

        //Returning String that holds time string
        return time;
    }
}
```

OUTPUT 6.23:

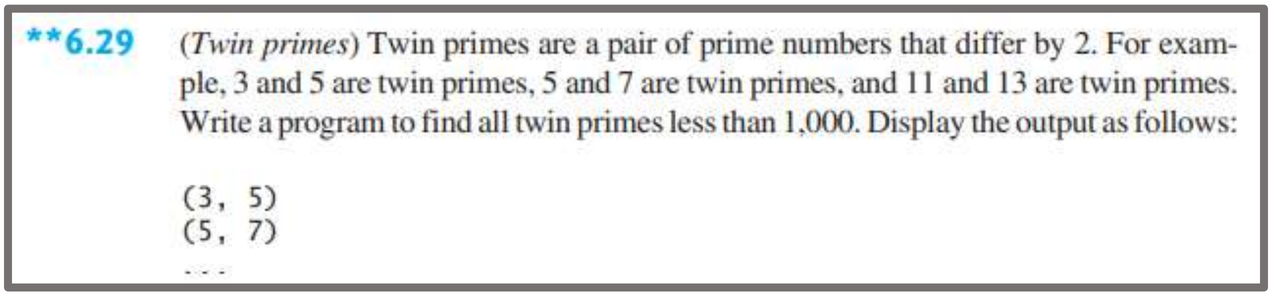


```
"C:\Program Files\Java\jdk-15.0
Enter Milli Seconds: 100000
Hours:Minutes:Seconds
0    :  1    :  40
```



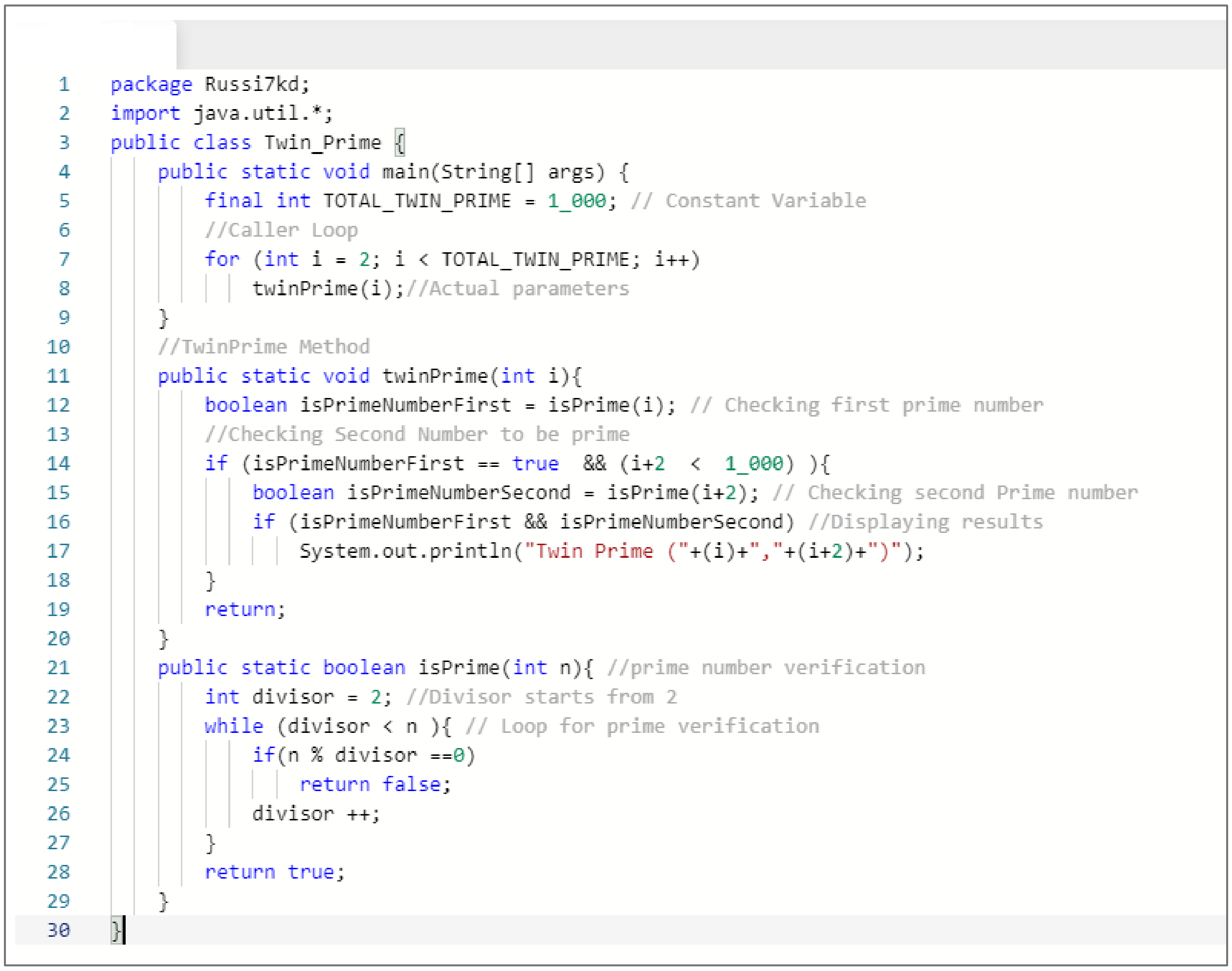
```
"C:\Program Files\Java\jdk-15.0.2\
Enter Milli Seconds: 555550000
Hours:Minutes:Seconds
154  :  19   :  10
```

Question 6.29



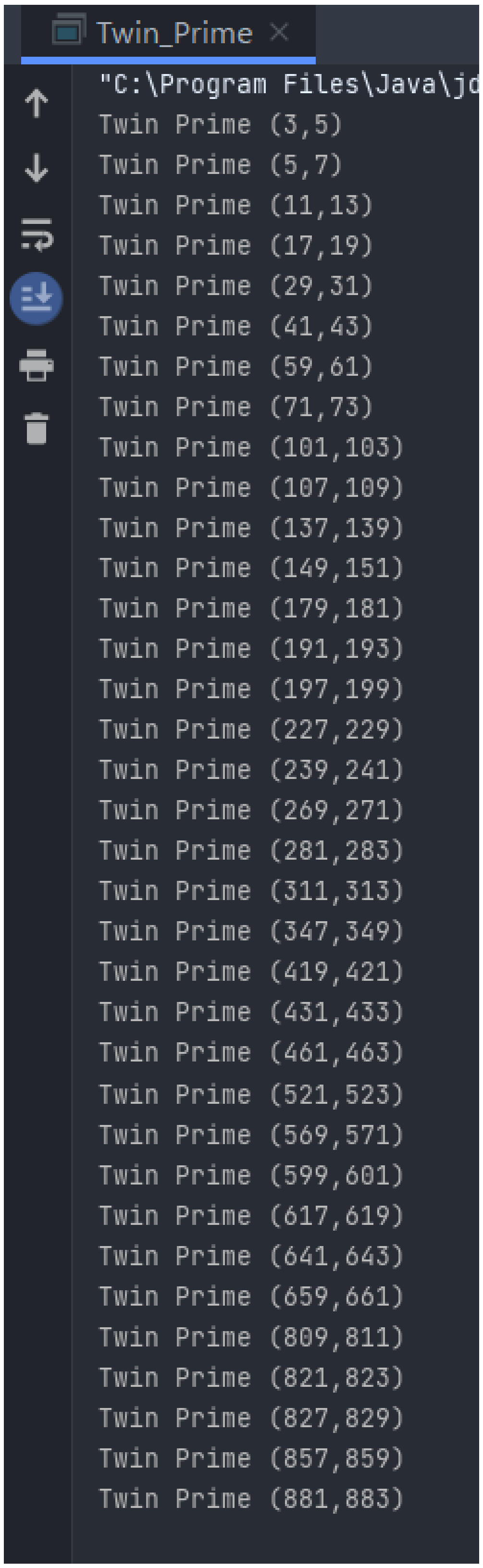
****6.29** (*Twin primes*) Twin primes are a pair of prime numbers that differ by 2. For example, 3 and 5 are twin primes, 5 and 7 are twin primes, and 11 and 13 are twin primes. Write a program to find all twin primes less than 1,000. Display the output as follows:

```
(3, 5)
(5, 7)
...
```



```java
package Russi7kd;
import java.util.*;
public class Twin_Prime {
    public static void main(String[] args) {
        final int TOTAL_TWIN_PRIME = 1_000; // Constant Variable
        //Caller Loop
        for (int i = 2; i < TOTAL_TWIN_PRIME; i++)
            twinPrime(i);//Actual parameters
    }
    //TwinPrime Method
    public static void twinPrime(int i){
        boolean isPrimeNumberFirst = isPrime(i); // Checking first prime number
        //Checking Second Number to be prime
        if (isPrimeNumberFirst == true  && (i+2  <  1_000) ){
            boolean isPrimeNumberSecond = isPrime(i+2); // Checking second Prime number
            if (isPrimeNumberFirst && isPrimeNumberSecond) //Displaying results
                System.out.println("Twin Prime ("+(i)+","+(i+2)+")");
        }
        return;
    }
    public static boolean isPrime(int n){ //prime number verification
        int divisor = 2; //Divisor starts from 2
        while (divisor < n ){ // Loop for prime verification
            if(n % divisor ==0)
                return false;
            divisor ++;
        }
        return true;
    }
}
```

OUTPUT 6.29:



```
"C:\Program Files\Java\jd
Twin Prime (3,5)
Twin Prime (5,7)
Twin Prime (11,13)
Twin Prime (17,19)
Twin Prime (29,31)
Twin Prime (41,43)
Twin Prime (59,61)
Twin Prime (71,73)
Twin Prime (101,103)
Twin Prime (107,109)
Twin Prime (137,139)
Twin Prime (149,151)
Twin Prime (179,181)
Twin Prime (191,193)
Twin Prime (197,199)
Twin Prime (227,229)
Twin Prime (239,241)
Twin Prime (269,271)
Twin Prime (281,283)
Twin Prime (311,313)
Twin Prime (347,349)
Twin Prime (419,421)
Twin Prime (431,433)
Twin Prime (461,463)
Twin Prime (521,523)
Twin Prime (569,571)
Twin Prime (599,601)
Twin Prime (617,619)
Twin Prime (641,643)
Twin Prime (659,661)
Twin Prime (809,811)
Twin Prime (821,823)
Twin Prime (827,829)
Twin Prime (857,859)
Twin Prime (881,883)
```