

PRACTICAL NO – 18

Write a java file handling program to count and display the number of palindromes present in a text file "myfile.txt". Example:

If the file "myfile.txt" contains the following lines,

My name is NITIN

Hello aaa and bbb word

How are You

ARORA is my friend

Output will be => 4

SOURCE CODE –

```
import java.io.*;
import java.util.*;

public class PalindromeCounter {

    public static void main(String[] args) throws IOException {

        int count = 0;

        FileReader fr = new FileReader("first.txt");
        BufferedReader br = new BufferedReader(fr);

        String line;

        System.out.println("Palindrome words are:");

        while ((line = br.readLine()) != null) {

            String[] words = line.split("\\s+");

            for (String word : words) {

                if (isPalindrome(word)) {

                    System.out.println(word);

                    count++;

                }

            }

        }

    }

}
```

```

        br.close();

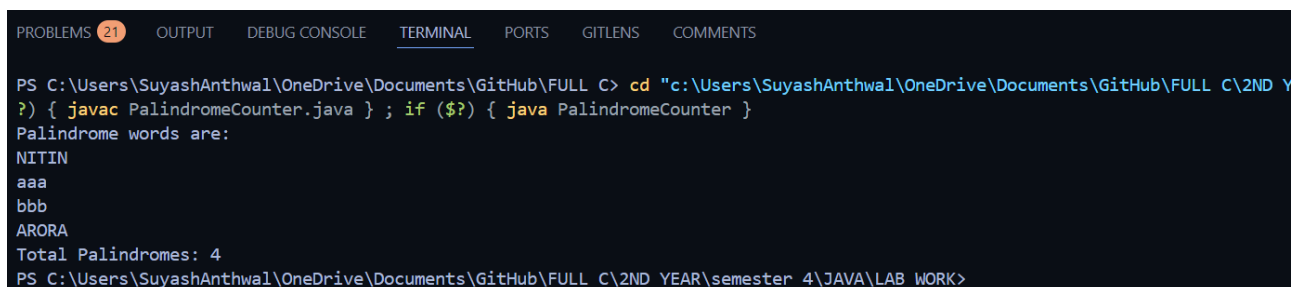
        System.out.println("Total Palindromes: " + count);
    }

    public static boolean isPalindrome(String word) {
        word = word.toLowerCase();
        int left = 0, right = word.length() - 1;

        while (left < right) {
            if (word.charAt(left) != word.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true ;
    }
}

```

OUTPUT-



```

PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C> cd "c:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C\2ND Y
?) { javac PalindromeCounter.java } ; if ($?) { java PalindromeCounter }
Palindrome words are:
NITIN
aaa
bbb
ARORA
Total Palindromes: 4
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C\2ND YEAR\semester 4\JAVA\LAB WORK>

```

PRACTICAL NO – 19

Write a program MultiThreads that creates two threads one thread with the name CSthread and the other thread named ITthread. Each thread should display its respective name and execute after a gap of 500 milliseconds. Each thread should also display a number indicating the number of times it got a chance to execute.

SOURCE CODE –

```
class CSthread extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            System.out.println(getName() + " executed " + i + " times");  
            try {  
                Thread.sleep(500);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
class ITThread extends Thread {  
    public void run() {  
        for (int i = 0; i < 5; i++) {  
            System.out.println(getName() + " executed " + i + " times");  
            try {  
                Thread.sleep(500);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```

    }
}

public class thread {

    public static void main(String[] args) {

        // Create and start the threads

        CSThread t1 = new CSThread();

        ITThread t2 = new ITThread();

        t1.start();

        t2.start();

    }

}

```

OUTPUT-



```

PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C> cd "c:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C\2ND YEAR\semester 4\JAVA\LAB WORK"
?) { javac thread.java } ; if ($?) { java thread }
Thread-1 executed 1 times
Thread-0 executed 1 times
Thread-1 executed 2 times
Thread-0 executed 2 times
Thread-0 executed 3 times
Thread-1 executed 3 times
Thread-0 executed 4 times
Thread-1 executed 4 times
Thread-0 executed 5 times
Thread-1 executed 5 times
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C\2ND YEAR\semester 4\JAVA\LAB WORK>

```

PRACTICAL NO – 20

Write a java program for to solve producer consumer problem in which a producer produce a value and consumer consume the value before producer generate the next value.

SOURCE CODE –

```
import java.util.* ;

class PC1 {

    LinkedList<Integer> list = new LinkedList<>();

    int capacity = 2;

    int value = 0;

    public void produce() throws Exception {

        while (true) {

            synchronized (this) {

                while (list.size() == capacity) {

                    wait();

                }

                System.out.println("Producer is going to produce..... " + value);

                list.add(value++);

                notifyAll();

                Thread.sleep(500);

            }

        }

    }

    public void consume() throws Exception {

        while (true) {

            synchronized (this) {

                while (list.size() == 0) {

                    wait();

                }

            }

        }

    }

}
```

```

        System.out.println("Consumer is going to consume ..... " + list.removeFirst());
        notifyAll();
        Thread.sleep(500);
    }
}
}
}
}

```

```

class pro extends Thread {
    PC1 p;
    public pro(PC1 p) {
        this.p = p;
    }
    public void run() {
        try {
            p.produce();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```

class con extends Thread {
    PC1 p;
    public con(PC1 p) {
        this.p = p;
    }
    public void run() {
        try {
            p.consume();
        } catch (Exception e) {

```

```

        System.out.println(e);
    }
}
}

public class proANDcon {

    public static void main(String[] args) {

        PC1 p = new PC1();

        pro t1 = new pro(p);

        con t2 = new con(p);

        t1.start();

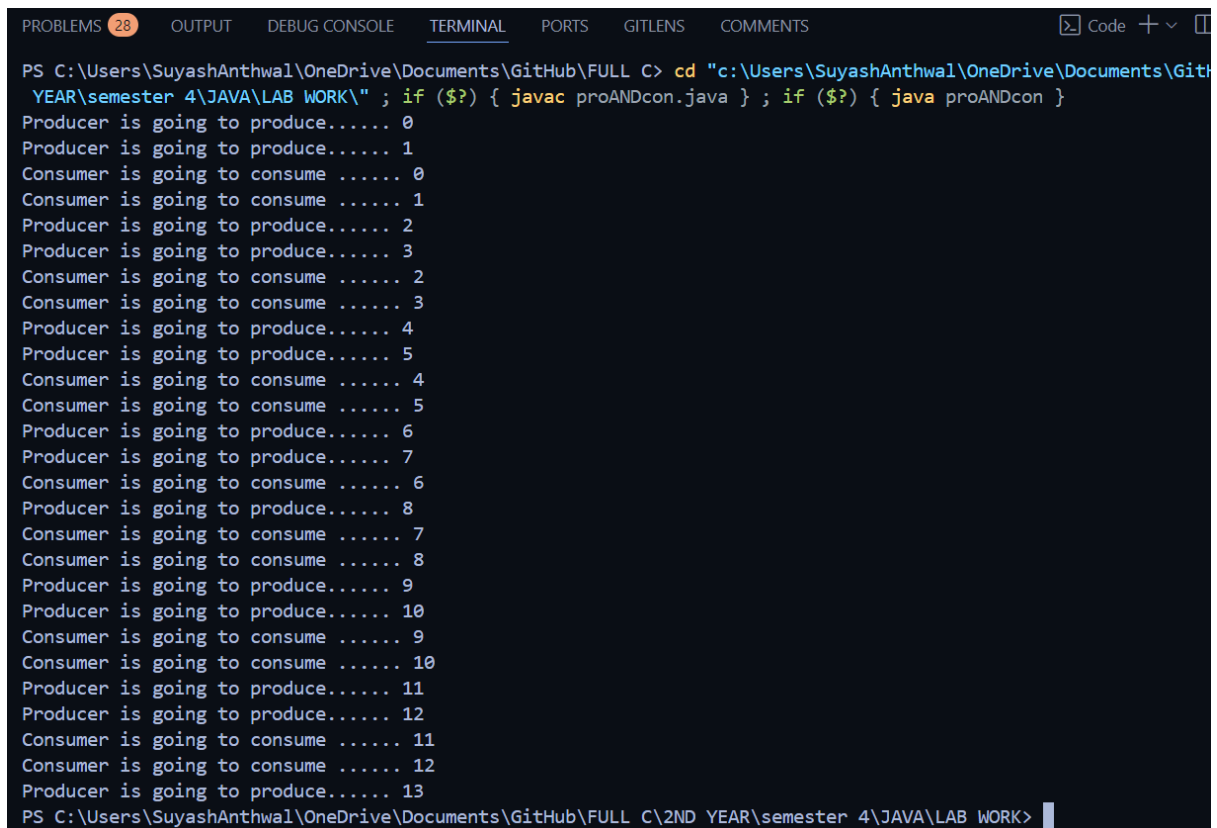
        t2.start();

    }

}

```

OUTPUT-



```

PROBLEMS 28 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS Code + - 
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C> cd "c:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\2ND YEAR\semester 4\JAVA\LAB WORK\" ; if ($?) { javac proANDcon.java } ; if ($?) { java proANDcon }
Producer is going to produce..... 0
Producer is going to produce..... 1
Consumer is going to consume ..... 0
Consumer is going to consume ..... 1
Producer is going to produce..... 2
Producer is going to produce..... 3
Consumer is going to consume ..... 2
Consumer is going to consume ..... 3
Producer is going to produce..... 4
Producer is going to produce..... 5
Consumer is going to consume ..... 4
Consumer is going to consume ..... 5
Producer is going to produce..... 6
Producer is going to produce..... 7
Consumer is going to consume ..... 6
Consumer is going to consume ..... 8
Producer is going to produce..... 8
Consumer is going to consume ..... 7
Consumer is going to consume ..... 8
Producer is going to produce..... 9
Producer is going to produce..... 10
Consumer is going to consume ..... 9
Consumer is going to consume ..... 10
Producer is going to produce..... 11
Producer is going to produce..... 12
Consumer is going to consume ..... 11
Consumer is going to consume ..... 12
Producer is going to produce..... 13
PS C:\Users\SuyashAnthwal\OneDrive\Documents\GitHub\FULL C\2ND YEAR\semester 4\JAVA\LAB WORK>

```