PARVATHAM RAM CHARAN

Assignment 10(IO STREAMS

---

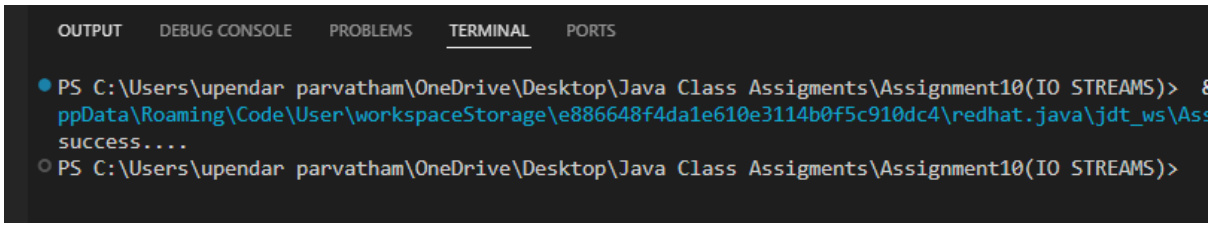
1. Write a Program to read the same program file and find the no. of lines, words and characters. Write the result in in to a text file (result.txt)

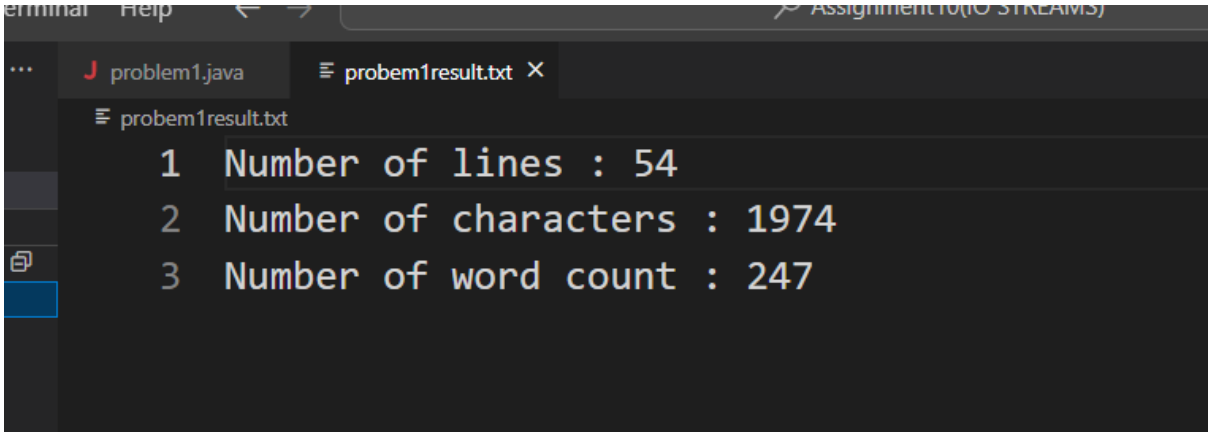
code:

```
import java.io.*;
public class problem1{
   public static void main(String[] args) throws Exception {
      FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//problem1.java");
      BufferedReader br = new BufferedReader(fr);
      String line = br.readLine();
      int lc=0,wc=0,cc=0;
      while(line !=null){
         lc++;
         cc += line.length();
         String[] words  =  line.trim().split("\\s+");
         //Splits the trimmed string into an array of words, using one or more
         //whitespace characters (\s+) as the delimiter.
         if(!line.trim().isEmpty()){
            wc +=words.length;
         }
         line=br.readLine();
      }
      fr.close();
      br.close();
      FileWriter fw = new FileWriter("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//probem1result.txt");
      BufferedWriter bw = new BufferedWriter(fw);
```

```
        bw.write("Number of lines : "+lc);
        bw.newLine();
        bw.write("Number of characters : "+cc);
        bw.newLine();
        bw.write("Number of word count : "+wc);
        bw.close();
        fw.close();
        System.out.println("success....");
    }
}

//\\s+ is a regular expression (regex) used in Java's split() method.
// 1. \s
// In regex, \s means any whitespace character.
// This includes:
// space (' ')
// tab (\t)
// newline (\n)
// carriage return (\r)
// form feed (\f)

// 2. +
// In regex, + means "one or more occurrences".
// So \s+ means one or more whitespace characters in a row.

// 3. \\s+ in Java
// In Java strings, \ is an escape character.
// So to represent the regex \s+, you must write \\s+ in Java.
// First \ escapes the second \ in the string.
// The regex engine finally sees \s+.
```

Output:





---

2. Write a program to read the same program file and write it to other file with

the lines number added before each line, starting from 1.

```
import java.io.*;

public class problem2 {

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

    FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//problem2.java");

    BufferedReader br = new BufferedReader(fr);


    FileWriter fw =  new FileWriter("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//problem2result.java");

    BufferedWriter bw = new BufferedWriter(fw);


    String line;

    int lc=1;
```

```
    while((line=br.readLine())!=null){
        fw.write((lc++)+" "+line+"\n");
    }
    fr.close();
    br.close();
    fw.close();
    bw.close();

    System.out.println("success");

    }
}
```

Output:

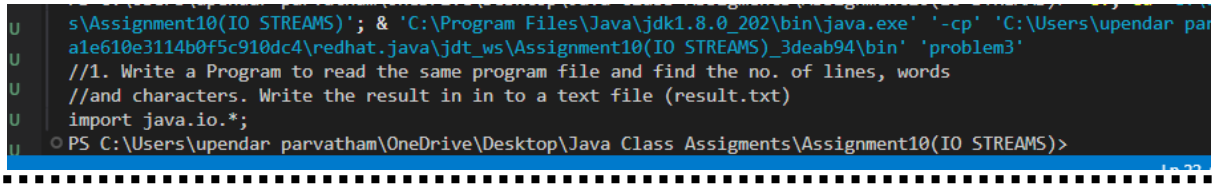
```
success
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\Assignment10(IO STREAMS)>
IO STREAMS)'; & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\upendar parva
java\jdt_ws\Assignment10(IO STREAMS)_3deab94\bin' 'problem2'
success
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\Assignment10(IO STREAMS)>
```

```java
//2. Write a program to read the same program file and write it to other file with
//the lines number added before each line, starting from 1.

import java.io.*;
public class problem2 {
    public static void main(String[] args) throws Exception{
    FileReader fr = new FileReader(fileName:"C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)/
    BufferedReader br = new BufferedReader(fr);

     FileWriter fw =  new FileWriter(fileName:"C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS
    BufferedWriter bw = new BufferedWriter(fw);

    String line;
    int lc=1;
    while((line=br.readLine())!=null){
        fw.write((lc++)+" "+line+"\n");
    }
    fr.close();
    br.close();
    fw.close();
    bw.close();

    System.out.println("success");

    }
}
```

3. Write a Java program to read first 3 lines from a file.

```
import java.io.BufferedReader;

import java.io.FileReader;

public class problem3 {

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

   FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//problem1.java");

   BufferedReader br = new BufferedReader(fr);


   String line="";

   int lc=0,max=3;

   while( ((line=br.readLine()) != null) &&   lc < max){

      System.out.println(line);

      lc++;

   }

   br.close();

   fr.close();


   }

}
```

Output:



4. Write a Java program to find the longest word in a text file.

```
import java.io.BufferedReader;

import java.io.FileReader;


public class problem4 {
```

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

   FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//sample.txt");

   BufferedReader br = new BufferedReader(fr);


   String maxword="";

   String word="";

   String line = "";

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

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

      for(String str : arr){

         str = str.replaceAll("[^a-zA-Z]","");

        if(maxword.length() < str.length()){

         maxword= str;

        }

      }

   }

   br.close();

   fr.close();

   System.out.println("The longest word in the file is: " +maxword);


   }

}
```

//[a-zA-Z] → matches all letters (lowercase a-z and uppercase A-Z)

//^ inside [] → negates the set, so [^a-zA-Z] means anything that is NOT a letter (numbers, punctuation, spaces, symbols).

//[^a-zA-Z] is a regex that matches any character that is not a letter (neither lowercase a–z nor uppercase A–Z).

nple.txt

```
1  Java is a powerful programming language
2  It is widely used for software development
3  Serialization, inheritance, polymorphism, encapsulation
4  Streams, files, and exception handling are important topics
5  Practice makes perfect in coding challenges
6
```

Output:

```
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\Assignment10(IO STREAMS)> &
ppData\Roaming\Code\User\workspaceStorage\e886648f4da1e610e3114b0f5c910dc4\redhat.java\jdt_ws\Assig
The longest word in the file is: Serialization
PS C:\Users\upendar parvatham\OneDrive\Desktop\Java Class Assigments\Assignment10(IO STREAMS)>
```

5.Write a programs to implemnt Caeser cipher using files.

Write to the file (enc_mssage.txt) with using caeser cipher with

the displacement value = 3.

Read the file (enc_message.txt) and decode the Cipher text and write it into

a file (dec_message.txt)

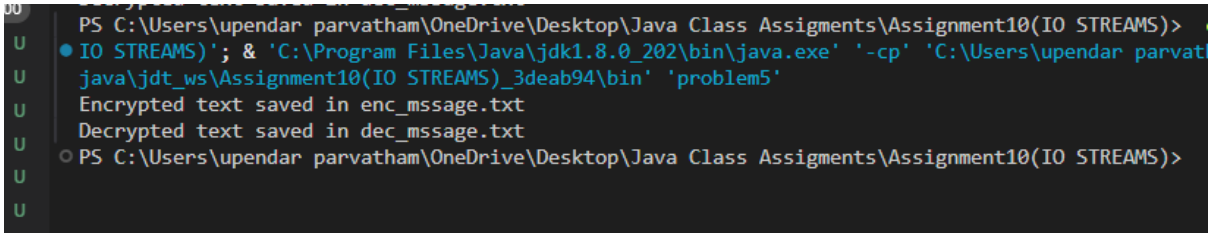
```
import java.io.*;

public class problem5 {

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

        FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//sample.txt");

        BufferedReader br = new BufferedReader(fr);


        int displacement = 3;

        StringBuilder original = new StringBuilder();

        String line;


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

            original.append(line).append('\n');

        }
```

```
// Encrypt the message
StringBuilder encrypt = new StringBuilder();
for (char ch : original.toString().toCharArray()) {
   if (Character.isUpperCase(ch)) {
      encrypt.append((char) ('A' + (ch - 'A' + displacement) % 26));
     //c = 'Z'
      //'Z' - 'A' = 25
      //25 + 3 = 28
      //28 % 26 = 2
      //'A' + 2 = 'C'

   } else if (Character.isLowerCase(ch)) {
      encrypt.append((char) ('a' + (ch - 'a' + displacement) % 26));
   } else {
      encrypt.append(ch); // keep spaces, punctuation, numbers as is
   }
}

FileWriter fw = new FileWriter("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//enc_mssage.txt");
BufferedWriter bw = new BufferedWriter(fw);
bw.write(encrypt.toString());
bw.close();
fw.close();
br.close();
fr.close();

System.out.println("Encrypted text saved in enc_mssage.txt");

// Decrypt
FileReader fr2 = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//enc_mssage.txt");
```

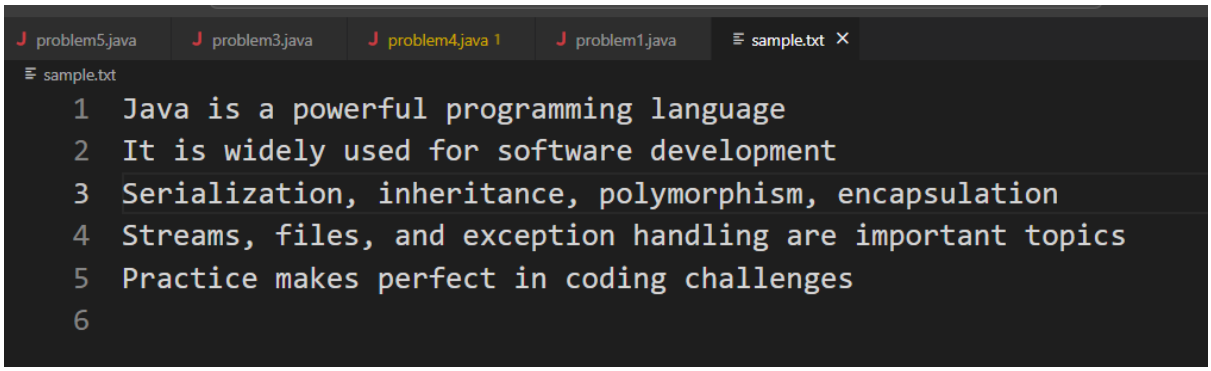
```
BufferedReader br2 = new BufferedReader(fr2);

StringBuilder encryptMessage = new StringBuilder();
while ((line = br2.readLine()) != null) {
   encryptMessage.append(line).append("\n");
}
br2.close();
fr2.close();

StringBuilder decrypt = new StringBuilder();
for (char ch : encryptMessage.toString().toCharArray()) {
   if (Character.isUpperCase(ch)) {
      decrypt.append((char) ('A' + (ch - 'A' - displacement + 26) % 26));
     //c = 'C'
      //'C' - 'A' = 2
      //2 - 3 = -1
      //-1 + 26 = 25
      //25 % 26 = 25
      //'A' + 25 = 'Z'

   } else if (Character.isLowerCase(ch)) {
      decrypt.append((char) ('a' + (ch - 'a' - displacement + 26) % 26));
   } else {
      decrypt.append(ch);
   }
}

FileWriter fw2 = new FileWriter("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//dec_mssage.txt");
BufferedWriter bw2 = new BufferedWriter(fw2);
bw2.write(decrypt.toString());
```

```
        bw2.close();
        fw2.close();

        System.out.println("Decrypted text saved in dec_mssage.txt");
    }
}
```

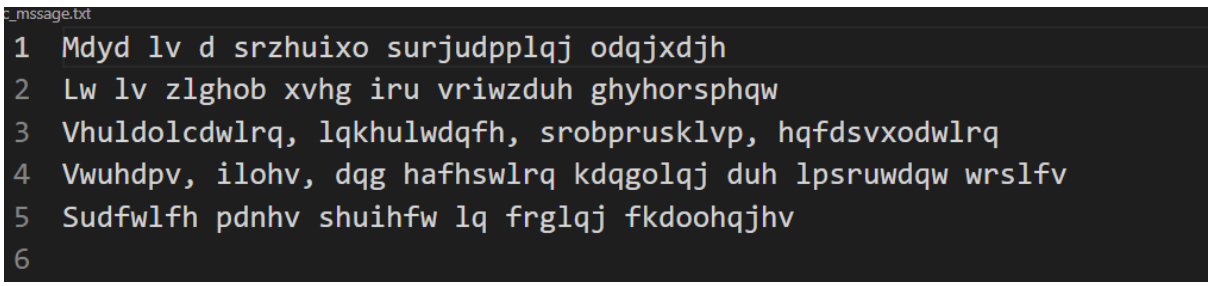
Output:



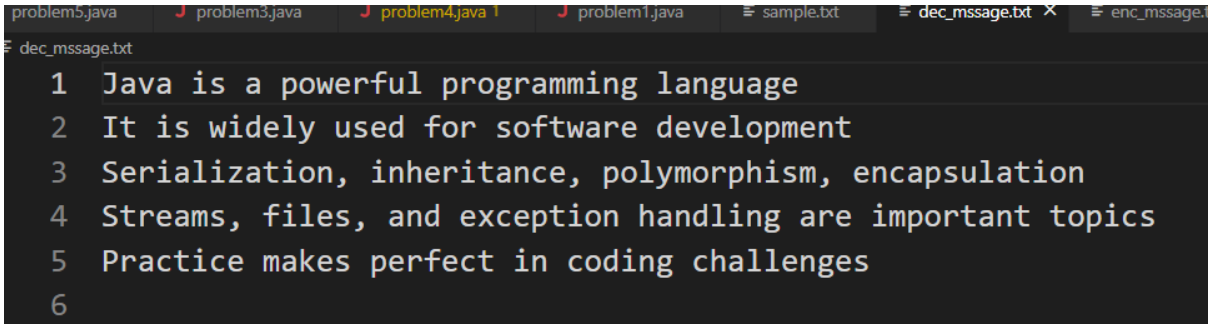
Sample.txt



Enc_msge.txt:



Dec_msge:



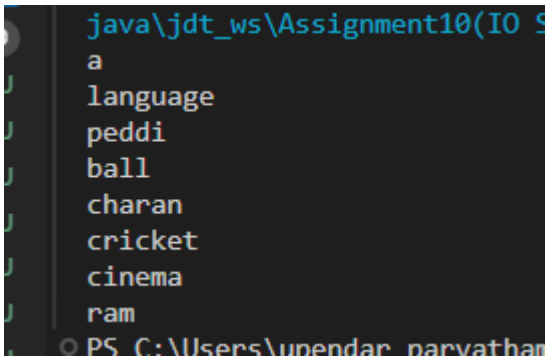
****

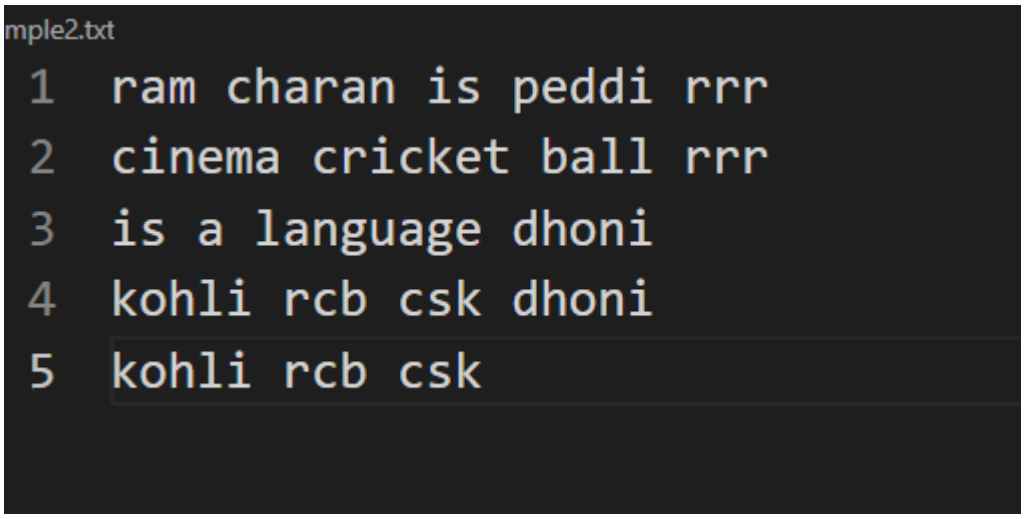
6.Write a program to find unique words in file

```
import java.io.*;
import java.util.HashMap;
import java.util.HashSet;
import java.util.*;
public class problem6 {
    public static void main(String[] args) throws Exception {
        FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//sample2.txt");
        BufferedReader br = new BufferedReader(fr);
        HashMap<String,Integer> hp = new HashMap<>();

        String line ="";
        while((line=br.readLine()) != null){
            String[] arr = line.trim().split("\\s+");
            for(String str: arr){
                hp.put(str,hp.getOrDefault(str,0)+1);
            }
        }
        for(Map.Entry<String,Integer> it : hp.entrySet()){
            if(it.getValue()==1){
                System.out.println(it.getKey());
            }
```

```
    }
  }
}
```

Output:





7. Write a program to find duplicate words in a file

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.HashMap;
import java.util.Map;

public class problem7 {
    public static void main(String[] args) throws Exception {
```

```
    FileReader fr = new FileReader("C://Users//upendar parvatham//OneDrive//Desktop//Java Class Assigments//Assignment10(IO STREAMS)//sample2.txt");
    BufferedReader br = new BufferedReader(fr);
    HashMap<String,Integer> hp = new HashMap<>();
    String line ="";
    while((line=br.readLine()) != null){
        String[] arr = line.trim().split("\\s+");
        for(String str: arr){
            hp.put(str,hp.getOrDefault(str,0)+1);
        }
    }
    for(Map.Entry<String,Integer> it : hp.entrySet()){
        if(it.getValue()>=2){
            System.out.println(it.getKey());
        }
    }
    }
}
```

Output: