Name: Pratyay Kumar
Date: 10/24/2022
Lab: 6
Lab Problem: File conversion

Program #6: File conversion -- remove control blocking data from backups

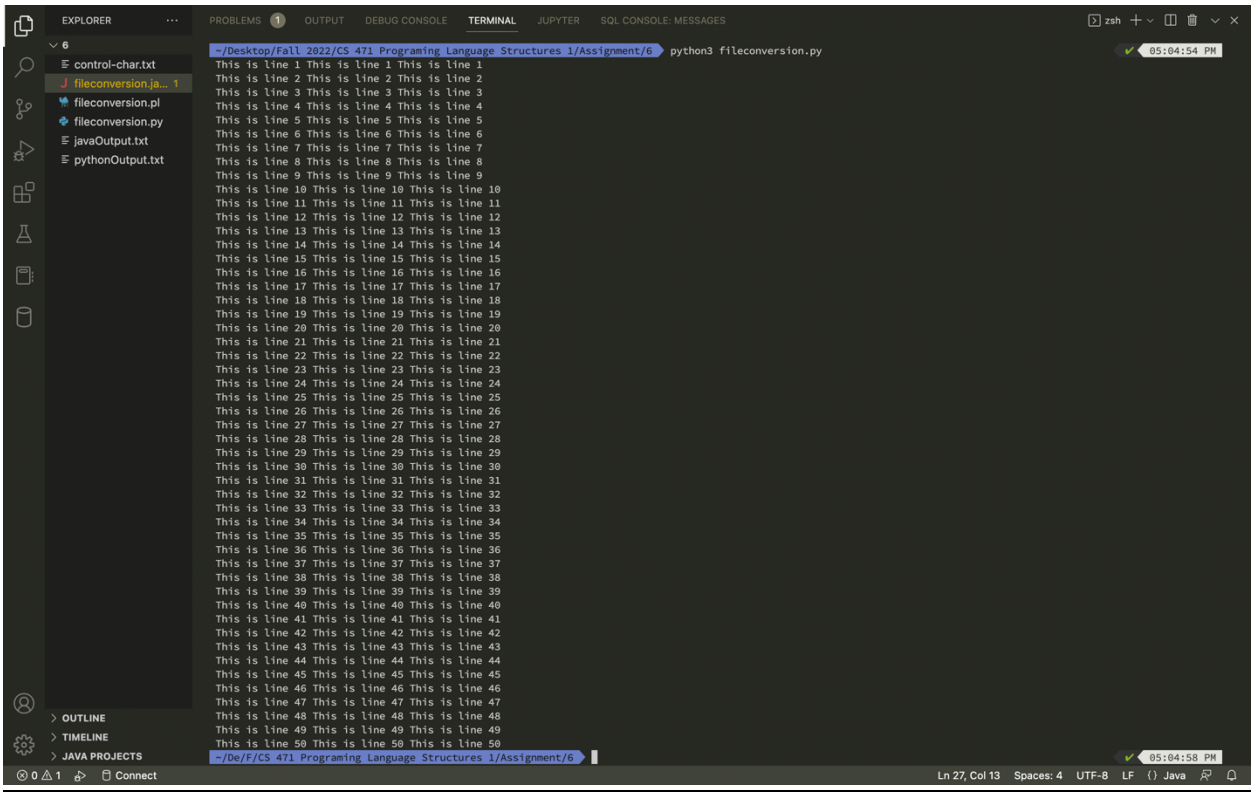
Problem Description: You are working on a project where you are given data that has been through a "backup" program which, at times, adds extraneous data. You discover that the data added has a pattern. The pattern is that the corrupt data ALWAYS starts with a CONTROL-C and always ends in a CONTROL-B, and that, at most, four carriage returns are introduced with such an event. You also discover that after the CONTROL-C, there may be more Control-C characters as part of the extraneous data.

Python Code:

```
# Author: Pratyay Kumar
# Date: 10/23/2022
# Lab: 6
# Program Description: File conversion -- removing control blocking data from backups
# Language: Python

import sys

def main():

    # Taking in the input file
    input = open("control-char.txt", 'r')

    # Storing in output file
    output = open('pythonOutput.txt', 'w')

    # Declaring variables to tell precisely if the control is inside ^C and outside ^B
    inC = False
    outC = False

    # Reading the input file
    line = input.read()

    # The for loop is used to convert characters to decimal value and
    # compater to ETX == 3 (^C) in decimal and STX == 2 (^B) in decimal.
    # The character between ETX and STX is not printed
    for i in line:
```

```
        if not line:
            break

        # Setting condition after ^C occurs
        if ord(i) == 3 and inC == False:
            inC = True
            outC = False

        # Getting conditions after ^B occurs
        if ord(i) == 2 and inC == True:
            inC = False
            outC = True

        # Printing the file, except the part in between ^C and ^B
        if inC == False and outC == False:
            # print(i, end = "")
            # output.write(i)
            sys.stdout.write(i)
            output.write(i)

        if outC == True:
            outC = False

    input.close()
    output.close()

if __name__ == "__main__":
    main()
```

The output of Python Code:



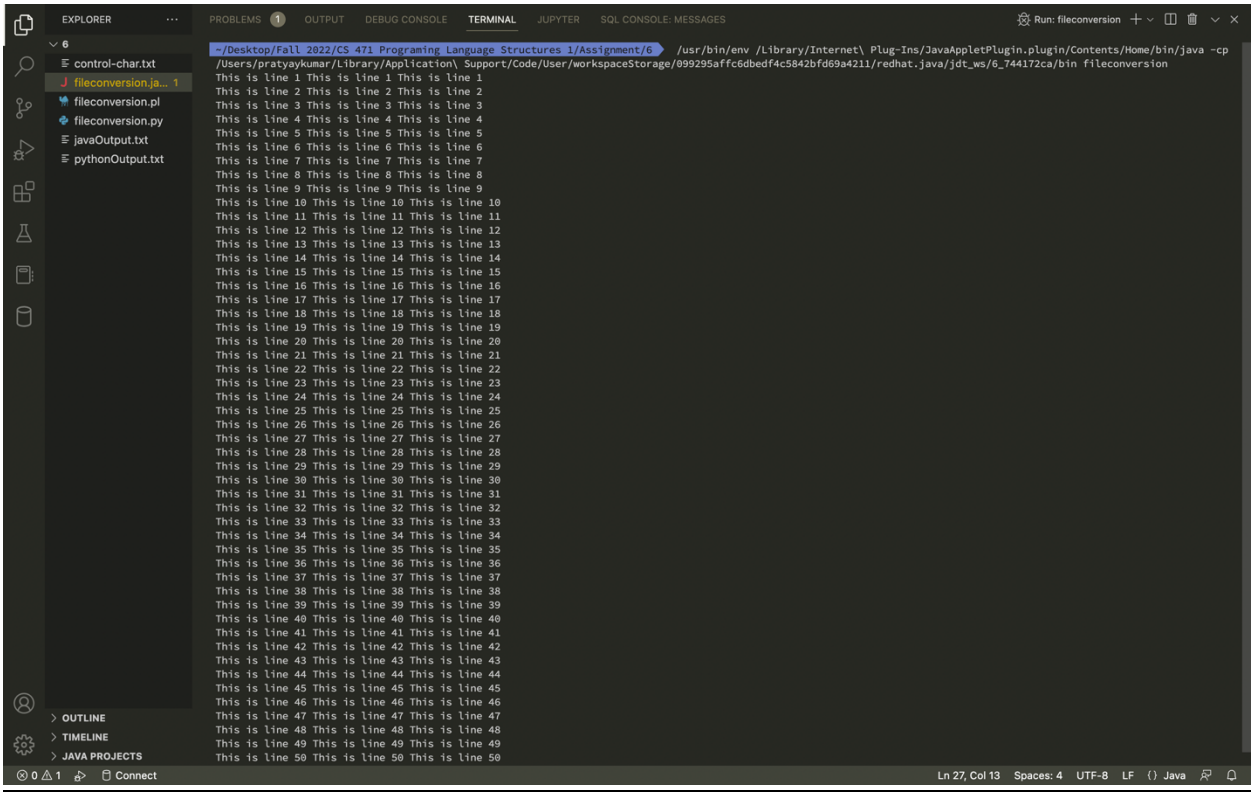
Java Code:

```java
/*
# Author: Pratyay Kumar
# Date: 10/23/2022
# Lab: 6
# Program Description: File conversion -- removing control blocking data from backups
# Language: Java
*/
import java.io.*;
import java.util.*;

public class fileconversion {
    public static void main( String[] args ) {

        try {
            // Taking in the input file
            FileReader input = new FileReader ("control-char.txt");
            Scanner s = new Scanner (input);

            // Storing in output file
            FileWriter output = new FileWriter ("javaOutput.txt");

            // Declaring variables to tell precisely if the control is inside ^C and
outside ^B
```

```
            boolean inC = false;
            boolean outC = false;

            String line;

            while ( s.hasNextLine() ) {

                // Reading the input file
                line = s.nextLine();
                int length = line.length();

                /*
                    The for loop is used to convert characters to decimal value and
                    compater to ETX == 3 (^C) in decimal and STX == 2 (^B) in decimal.
                    The character between ETX and STX is not printed
                */
                for ( int i=0; i<length; i++ ) {

                    // Setting condition after ^C occurs
                    if ( (int)line.charAt(i) == 3 && inC == false ) {
                        inC = true;
                        outC = false;
                    }
                    // Getting conditions after ^B occurs
                    if ( (int)line.charAt(i) == 2 && inC == true ) {
                        inC = false;
                        outC = true;
                    }
                    // Printing the file, except the part in between ^C and ^B
                    if ( inC == false && outC == false ) {
                        System.out.print(line.charAt(i));
                        output.write(line.charAt(i));
                    }

                    if ( outC == true ) {
                        outC = false;
                    }

                }

                if (inC == false) {
                    System.out.println();
                    output.write("\n");
                }

            }

            input.close();
```

```
            output.close();

        } catch ( IOException e ) {
            e.printStackTrace();
            System.exit(1);
        }
    }

}
```

The output of Java code:



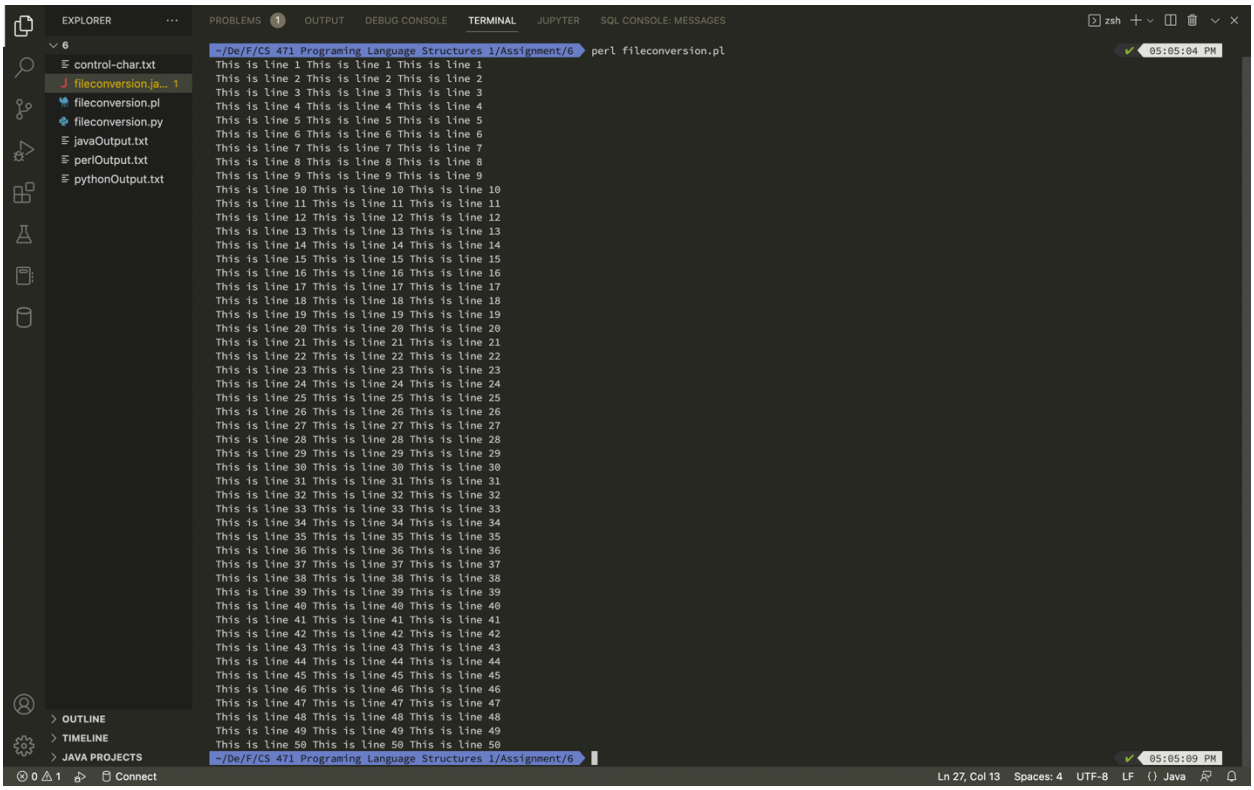
Perl Code:

```
# Author: Pratyay Kumar
# Date: 10/23/2022
# Lab: 6
# Program Description: File conversion -- removing control blocking data from backups
# Language: Perl
# !/usr/bin/perl

use strict;
use warnings;

# Taking in the input file
```

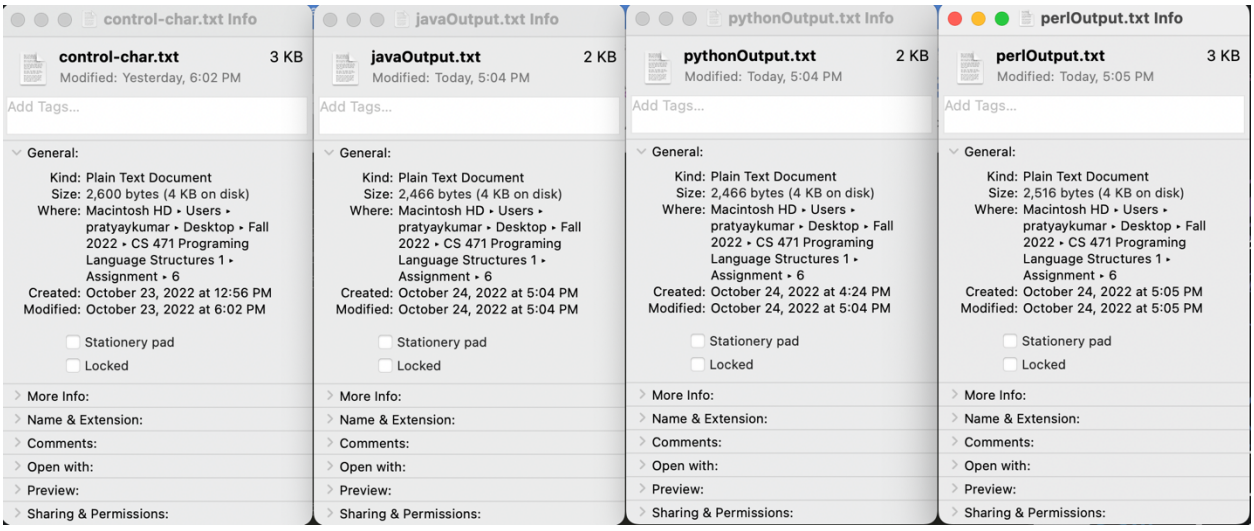
```perl
my $input = 'control-char.txt';
open(FH, '<', $input) or die $!;

# Storing in output file
open (FW, '>', "perlOutput.txt") or die $!;


# Declaring variables to tell precisely if the control is inside ^C and outside ^B
my $inC = 1; # False
my $outC = 1; # False

# Reading the input file
my $line = <FH>;

# The for loop is used to convert characters to decimal value and
# compater to ETX == 3 (^C) in decimal and STX == 2 (^B) in decimal.
# The character between ETX and STX is not printed
while ( $line ) {

    my $str = $line;
    # using split() function
    my @spl = split('', $str);

    # displaying result using foreach loop
    foreach my $i (@spl) {
        if (ord($i) == 3 and $inC == 1) {
            $inC = 0;
            $outC = 1;
        }
        # Getting conditions after ^B occurs
        if (ord($i) == 2 and $inC == 0) {
            $inC = 1;
            $outC = 0;
        }
        # Printing the file, except the part in between ^C and ^B
        if ($inC == 1 and $outC == 1) {
            print $i;
            $a = $i;
            print FW $a;
        }
        if ($outC == 0) {
            $outC = 1;
        }
    }

    $line = <FH>;

}
```

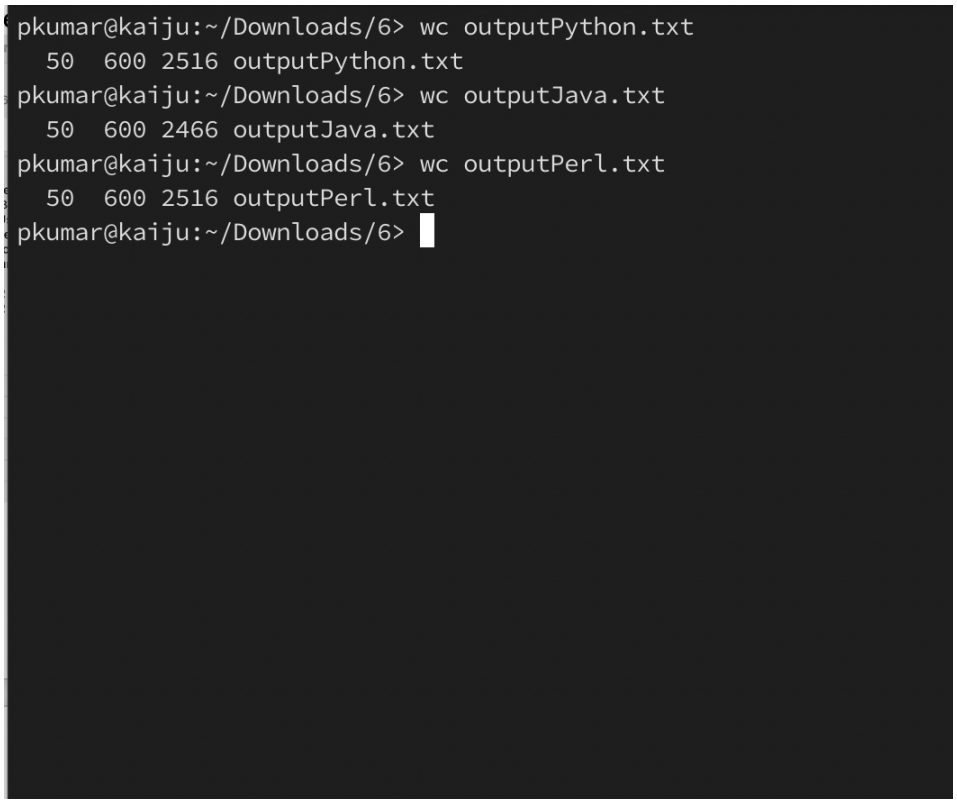
```
close(FW) or die $!;
close (FH) or die $!;
```

Output of Perl Code:



Conclusion of the file generated by the Python, Java, and Perl code: The file size for the output file generated is 2KB for Java, 2KB for Python, and 3KB for Perl. You can see the screenshot below.





The above screenshot shows the word counts of the output file generated by the codes.