Tony Maldonado
CS 471
November 1, 2020

Program 7: File Conversion – Remove Control Blocking data from Backups

Problem Description:

- We are given a text file which contains added on corrupt data. This data was added on at the time that the file was backed up using some backup program. The corrupt data that was added on is the control blocking characters ‘CONTROL C’ and ‘CONTROL B’, and on top of this, carriage returns were added into the file. We were then to write programs in Perl, Python, and Java which remove these values from the file.
- In all three language implementations I read the file and used a flag to determine whether to print the current looked at character to the output file or not. I set the flag initially to true and then once I reached a CTRL C I set to false in which I stopped printing the characters until I reached a CTRL B where I set the flag to true and allowed printing to resume. This method proved to be successful in all three languages.

Tony Maldonado
CS 471
November 1, 2020

Perl Code & Output:

```
#!/usr/bin/perl

# Name: Tony Maldonado
# Date: October 31, 2020
# Input: A txt file which has corrupt data in it.
# Output: A new txt file with the corrupt data removed.
# Preconditions: The txt file should exist and be named 'control-char.txt'.
# Postconditions: The file is wanted with only printable characters.
#                                 e.g., no control blocking chars and no carriage return
use strict;
use warnings;

# Variable declarations:
# These are the input and output files
my $inputFile = "control-char.txt";
my $outputFile = "newPerlFile.txt";
# String will store the final string
my $fileString = "";
# Flag to determine whether we should print
my $flag = 1;
# Read and char are for the file reader
my $read;
my $char;

# Open the file in read mode
open FH, '<', $inputFile or die "No such file, can't open! $!";

# Use an integer flag to determine what gets printed to the new file.
# Initially flag is true, so it starts off printing fine. Once it
# encounters a CTRL C it will set the flag to False and not print until
# we encounter a CTRL B and set the flag to true afterwords
while ($read = read FH, $char, 1) {
        # CTRL C sets flag to 0 and doesn't print
        if (ord($char) == 3) {
                $flag = 0;
        }
        # CTRL B sets flag to 1
        if (ord($char) == 2) {
                $flag = 1;
        }
        # Skip to next iteration
        next if (ord($char) == 2);
        if ($flag ==1){
                $fileString .= $char;
        }
}
```

Tony Maldonado
CS 471
November 1, 2020

```
# Create the file for the output and print to it
open FH, '>', $outputFile or die "No such file, can't open! $!";
print FH $fileString;
print "New file created successfully: $outputFile\n";
```

```
Tonys-MacBook-Pro:Program 7 m21tony$ perl controlRemoval.pl
New file created successfully: newPerlFile.txt
Tonys-MacBook-Pro:Program 7 m21tony$
```

Program 7 — vim newPerlFile.txt — 78×53

```
This is line 1 This is line 1 This is line 1
This is line 2 This is line 2 This is line 2
This is line 3 This is line 3 This is line 3 A
This is line 4 This is line 4 This is line 4
This is line 5 This is line 5 This is line 5
This is line 6 This is line 6 This is line 6
This is line 7 This is line 7 This is line 7
This  is line 8 This is line 8 This is line 8 A
This is line 9 This is line 9 This is line 9
This is line 10 This is line 10 This is line 10
This is line 11 This is line 11 This is line 11
This is line 12 This is line 12 This is line 12
This is line 13 This is line 13 This is line 13
This is line 14 This is line 14 This is line 14
This is line 15 This  is line 15 This is line 15 B
This is line 16 This is line 16 This is line 16
This is line 17 This is line 17 This is line 17
This is line 18 This is line 18 This is line 18
This is line 19 This is line 19 This is line 19
This is line 20 This is line 20 This is line 20 B
This is line 21 This is line 21 This is line 21
This is line 22 This is line 22 This is line 22
This is line 23 This is line 23 This is line 23
This is line 24 This is line 24 This is line 24
This is line 25 This is line 25 This is line 25 B
This is line 26 This is line 26 This is line 26
This is line 27 This is line 27 This is line 27
This is line 28 This is line 28 This is line 28
This is line 29 This is line 29 This is line 29
This is line 30 This is line 30 This is line 30
This is line 31 This is line 31 This is line 31
This is line 32 This is line 32 This is line 32
This is line 33 This is line 33 This is line 33
This is line 34 This is line 34 This is line 34
This is line 35 This is line 35 This is line 35
This is line 36 This is line 36 This is line 36
This is line 37 This is line 37 This is line 37
This is line 38 This is line 38 This is line 38
This is line 39 This is line 39 This is line 39
This is line 40 This is line 40 This is line 40
This is line 41 This is line 41 This is line 41
This is line 42 This is line 42 This is line 42
This is line 43 This is line 43 This is line 43
This is line 44 This is line 44 This is line 44
This is line 45 This is line 45 This is line 45
This is line 46 This is line 46 This is line 46
This is line 47 This is line 47 This is line 47
This is line 48 This is line 48 This is line 48
This is line 49 This is line 49 This is line 49  C
This is line 50 This is line 50 This is line 50 D
~
~
```

Tony Maldonado
CS 471
November 1, 2020

Java Code & Output:

```
/*
* Name: Tony Maldonado
* Date: November 01, 2020
* Input: A txt file which has corrupt data in it
* Output: A new txt file with the corrupt data removed.
* Preconditions: The text file should exist and be named 'control-char.txt'.
* Postconditions: The file is wanted with everything in between CTRL C and
*                                 CTRL B removed. All other characters stay in place.
*/

/* Sources:
*       https://stackoverflow.com/questions/811851/how-do-i-read-input-character-by-character-in-java
*       https://www.baeldung.com/java-write-to-file
*       https://stackoverflow.com/questions/5868369/how-can-i-read-a-large-text-file-line-by-line-using-java
*/

import java.io.*;
import java.util.*;
import java.nio.file.*;

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

                // First read the input file stream
                FileInputStream fstream = new FileInputStream("control-char.txt");
                BufferedReader br = new BufferedReader(new InputStreamReader(fstream));

                // Then create the new output file
                BufferedWriter writer = new BufferedWriter(new FileWriter("newJavaFile.txt"));

                // Use an integer flag to determine what gets printed to the new file.
                // Initially flag is true, so it starts off printing fine. Once it
                // encounters a CTRL C it will set the flag to False and not print until
                // we encounter a CTRL B and set the flag to true afterwords
                int flag = 1;
                int r;

                while ((r = br.read()) != -1) {
                        char ch = (char) r;
                        if (ch == 0x3)
                                flag = 0;
                        if (ch == 0x2){
                                flag = 1;
```

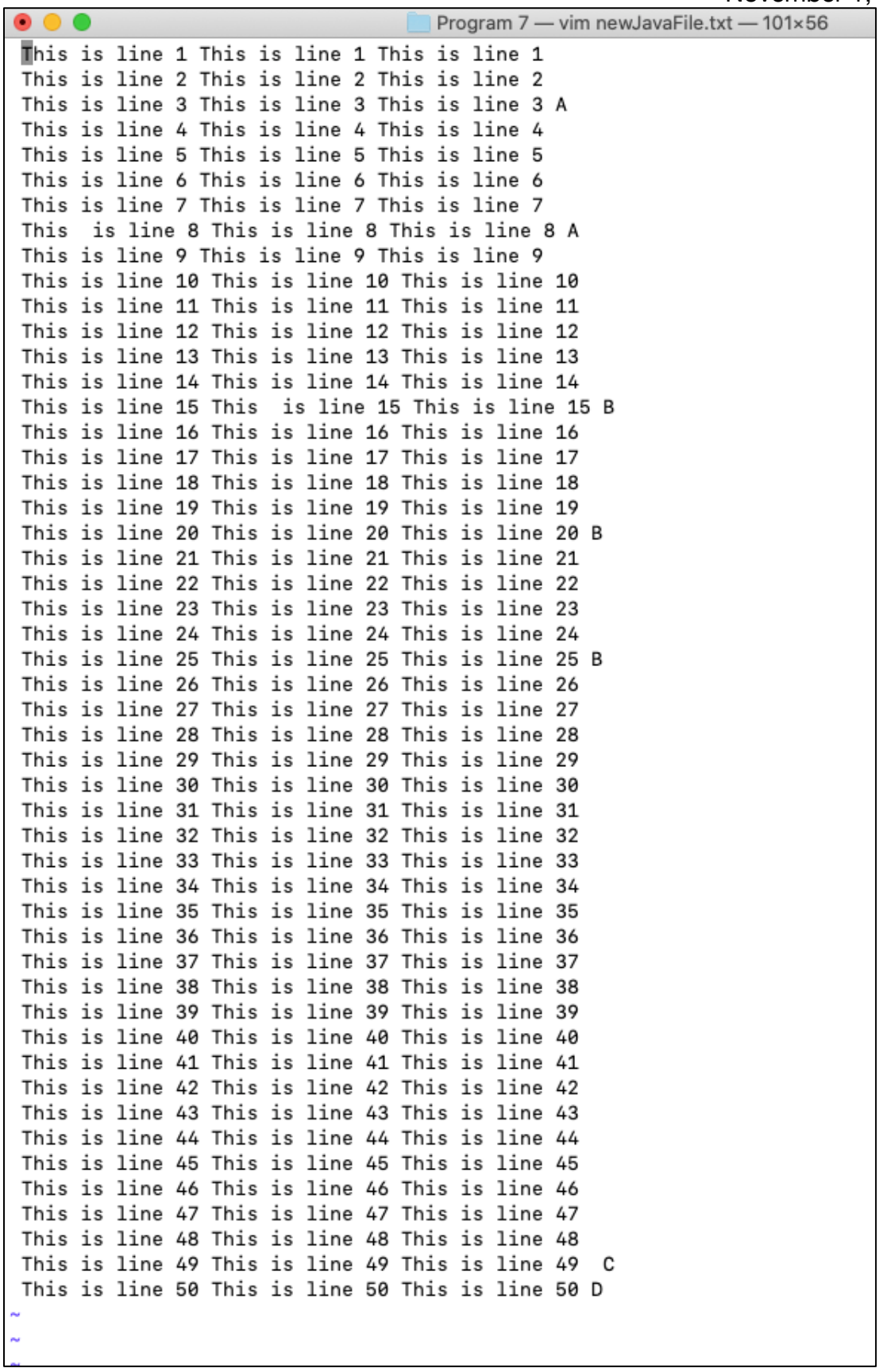
```
                        continue;
                }
                if (flag == 1)
                        writer.write(ch);
        }

        System.out.println("New file successfully created: newJavaFile.txt");

        /// Close both files
        fstream.close();
        writer.close();
    }
}
```

```
Tonys-MacBook-Pro:Program 7 m21tony$ javac controlRemoval.java
Tonys-MacBook-Pro:Program 7 m21tony$ java controlRemoval.java
New file successfully created: newJavaFile.txt
Tonys-MacBook-Pro:Program 7 m21tony$
```

Tony Maldonado
CS 471
November 1, 2020



Tony Maldonado
CS 471
November 1, 2020

Python Code & Output:

```
# Name: Tony Maldonado
# Date: October 31, 2020
# Input: A txt file which has corrupt data in it.
# Output: A new txt file with the corrupt data removed.
# Precondition: The txt file should exist and be named 'control-char.txt'.
# Postcondition: The file is wanted with everything in between CTRL C and
#                                 CTRL B removed. All other characters stay in place.

import string

# Open the txt file and assign it to a string
file1 = open("control-char.txt")

# New file for writing the modified contents to
file2 = open("newPythonFile.txt", "w")

# Use a boolean flag to determine what gets printed to the new file.
# Initially flag is true, so it starts off printing fine. Once it
# encounters a CTRL C it will set the flag to False and not print until
# we encounter a CTRL B and set the flag to true afterwords
flag = True

for lines in file1:
        for char in lines:
                if char == chr(3):
                        flag = False
                if char == chr(2):
                        flag = True
                        continue
                if flag == True:
                        file2.write(char)

# New modified file created successfully
print("New file created successfully: ")
print(file2.name)

# Close both files
file1.close()
file2.close()
```

Tony Maldonado
CS 471
November 1, 2020

```
Tonys-MacBook-Pro:Program 7 m21tony$ python3 controlRemoval.py
New file created successfully:
newPythonFile.txt
```

Program 7 — vim newPythonFile.txt — 101×56

```
This is line 1 This is line 1 This is line 1
This is line 2 This is line 2 This is line 2
This is line 3 This is line 3 This is line 3 A
This is line 4 This is line 4 This is line 4
This is line 5 This is line 5 This is line 5
This is line 6 This is line 6 This is line 6
This is line 7 This is line 7 This is line 7
This  is line 8 This is line 8 This is line 8 A
This is line 9 This is line 9 This is line 9
This is line 10 This is line 10 This is line 10
This is line 11 This is line 11 This is line 11
This is line 12 This is line 12 This is line 12
This is line 13 This is line 13 This is line 13
This is line 14 This is line 14 This is line 14
This is line 15 This  is line 15 This is line 15 B
This is line 16 This is line 16 This is line 16
This is line 17 This is line 17 This is line 17
This is line 18 This is line 18 This is line 18
This is line 19 This is line 19 This is line 19
This is line 20 This is line 20 This is line 20 B
This is line 21 This is line 21 This is line 21
This is line 22 This is line 22 This is line 22
This is line 23 This is line 23 This is line 23
This is line 24 This is line 24 This is line 24
This is line 25 This is line 25 This is line 25 B
This is line 26 This is line 26 This is line 26
This is line 27 This is line 27 This is line 27
This is line 28 This is line 28 This is line 28
This is line 29 This is line 29 This is line 29
This is line 30 This is line 30 This is line 30
This is line 31 This is line 31 This is line 31
This is line 32 This is line 32 This is line 32
This is line 33 This is line 33 This is line 33
This is line 34 This is line 34 This is line 34
This is line 35 This is line 35 This is line 35
This is line 36 This is line 36 This is line 36
This is line 37 This is line 37 This is line 37
This is line 38 This is line 38 This is line 38
This is line 39 This is line 39 This is line 39
This is line 40 This is line 40 This is line 40
This is line 41 This is line 41 This is line 41
This is line 42 This is line 42 This is line 42
This is line 43 This is line 43 This is line 43
This is line 44 This is line 44 This is line 44
This is line 45 This is line 45 This is line 45
This is line 46 This is line 46 This is line 46
This is line 47 This is line 47 This is line 47
This is line 48 This is line 48 This is line 48
This is line 49 This is line 49 This is line 49  C
This is line 50 This is line 50 This is line 50 D
~
~
~
```