

## Lab 1 Part 4

1)

Write a C program (lab1p4\_p1.c) that reads an input text file, words.txt (at most 500 words; words may include special characters and numbers).

Your program should then count the distinct words (not case sensitive and space-delimited) in the file and save the number of distinct words in out.txt.

lab1p4\_1.c

For this part I requested the user to input a file location/name. Then I opened the file, and checked if the input file and output files opened correctly. I then used a while loop conditioned on the code reading characters from the file. This loop repeated until the end of the file. For part 1, if the character read was a space, new line or a tab, I filled out the rest of the array length with null characters and moved to the next index. I then iterated through the array and compared the words stored in the array to each other. If a match only occurred once (the word matched with itself) I increased the number of distinct words. I then added a value to my count, because I found that my algorithm was always one number less than the accurate value. I then wrote the number of distinct terms to the file "out.txt".

The screenshot displays a C program running in a terminal window. The program is processing a file named 'in1.txt' and writing the result to 'out.txt'. The output shows that there are 8 distinct words in the input file. The program also displays a warning about the use of the unsafe 'gets()' function and ends with an exit code of 0.

```
in1.txt
You cannot end a sentence with because because. because is a conjunction

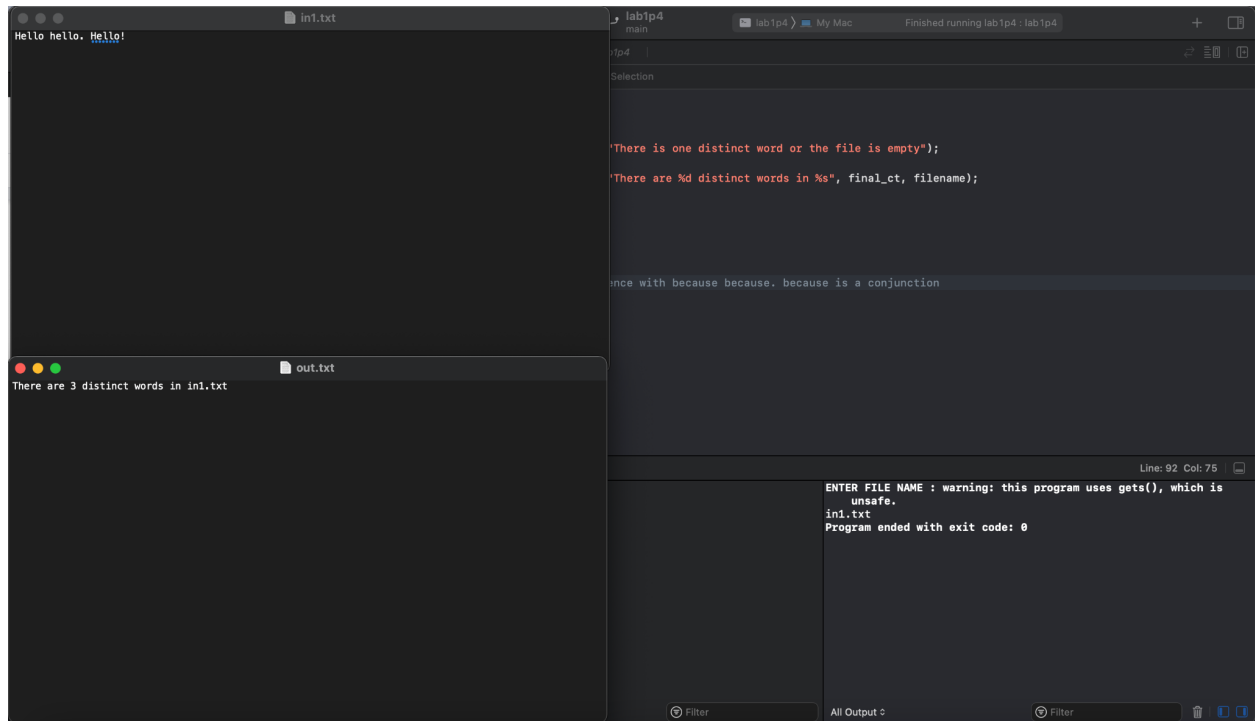
out.txt
There are 8 distinct words in in1.txt

lab1p4
main
lab1p4 > My Mac
Finished running lab1p4 : lab1p4

ip4
main()

"There is one distinct word or the file is empty");
"There are %d distinct words in %s", final_ct, filename);

ENTER FILE NAME : warning: this program uses gets(), which is
unsafe.
in1.txt
Program ended with exit code: 0
```



2)

Write a C program (lab1p4\_p2.c) that improves on p1, to recognize "file" and "file." as the same word "file".

For this part I requested the user to input a file location/name. Then I opened the file, and checked if the input file and output files opened correctly. I then used a while loop conditioned on the code reading characters from the file. This loop repeated until the end of the file. For part 2, if the ascii value was outside of the values for letters I filled out the rest of the array length with null characters and moved to the next index. I then iterated through the array and compared the words stored in the array to each other. If a match only occurred once (the word matched with itself) I increased the number of distinct words. I then added a value to my count, because I found that my algorithm was always one number less than the accurate value. I then wrote the number of distinct terms to the file "out.txt".

