

Matric No.: 222499

Name: Oladokun Tosin Emmanuel

<https://github.com/OLADISEA/School-project-3>

step 1: start

step 2: Declare numTestCase to store the number of test case.

step 3: Declare a struct file named File which will store two variables called fileName and Id

step 4: Declare a file pointer called file to read the .in file.

step 5: Read the .in file into the file pointer variable.

step 6: if(file == Null), Display "Error accessing the file and go to 50

step 7: Declare and initialize count to 0.

step 8: if(count > numTestCase), then go to 50

step 9: Declare the struct variable called file1.

step 10: Declare the numFiles to store the number of files in each test case.

step 11: Read the number from the .in file and store it in numFiles.

step 12: Declare and initialize i to 0 to loop through the numFiles.

step 13: if(i > numFiles), then go to step 49

step 14: Read the fileName and id from the .in file.

step 15: store them inside the variable created in the struct file(file1) declared in step 8

step 16: The fileName and id must be stored from index i to the numFiles - 1

step 17: increment i and go to step 13

step 18: Declare temp,temp2 to temporarily store the fileName and id respectively.

step 19: Declare i,len and initialize len to numFiles - 1 and let i be initialized to 0

step 20: if(i > len), then go to 30

step 21: Declare and initialize j to i + 1.

step 22: if(j > numFiles), then go to 29

step 23: Declare and initialize compare to compare the index of i fileName and the index of i in id

step 24: swapping is needed to be done if some conditions are true to make it in ascending order.

step 25: if(compare == 0), set temp2 to the value of i in file1.id, set the value of i in file.id to be equal to the value of j in file.id and lastly set the value of j in file.id to be equal to temp2. This will push the smallest id of the same file to the front, then go to step 28.

step 26: copy the value of i in file1.fileName into temp, copy the value of i in file.fileName to be equal to the value of j in file.fileName and lastly copy the value of j in file.fileName to be equal to temp2. This will push the smallest fileName the front, then go to step 28

step 27: if(compare > 0), repeat step 25 and step 26

step 28: increment j by 1, then go to step 22

step 29: increment i by 1, then go to step 20

step 30: create a loop to remove the duplicate files.

step 31: Declare and initialize i to 0.

step 32: if(i >= numFiles), go to step 44

step 33: Declare and initialize j to i + 1

step 34: if(j >= numFiles), go to step 43

step 35: if the value of i in file1.fileName and the value of j in file1.fileName are equal, then

step 36: Declare and initialize k to j.

step 37: if(k >= numFiles), go to step 41

step 38: if the value of k in file1.id is not equal to the value of k+1 in file1.id, go to step 42

step 39: copy the value of k+1 in file1.fileName to the value of k in file1.fileName.

step 40: increment k by 1, then go to step 37

step 41: decrement the value of numFiles by 1

step 42: increment j by 1, then go to step 34

step 43: increment i by 1, then go to step 32

step 44: Declare and initialize i to 0

step 45: if(i >= numFiles), then go to step 50

step 46: Display the value of i in file1.id

step 47: Increment i by 1, then go to step 45

step 48: Increment i by 1, then go to step 13

step 49: Increment count by 1, then go to step 8

step 50: close the file

step 51: stop