**Explanation For combining char alternatively from Two Different file**

#!/bin/bash

This line specifies that the script should be interpreted using the Bash shell.

# Prompt the user to enter the input file names

read -p "Enter the path first input file name: " file1

read -p "Enter the path second input file name: " file2

read -p "Enter the path output file name: " output\_file

These lines prompt the user to enter the names of two input files (that previously created by split char ) and an output file(that we want with extension). The user's inputs are stored in the variables file1, file2, and output\_file, respectively.

# Check if the input files exist

if [ ! -f "$file1" ]; then

echo "File '$file1' does not exist."

exit 1

fi

if [ ! -f "$file2" ]; then

echo "File '$file2' does not exist."

exit 1

Fi

These lines check if the input files specified by the user exist. If any of the files does not exist (! -f checks if the file does not exist or is not a regular file), an error message is displayed, and the script exits with an exit code of 1.

# Read the characters from each file

chars1=$(cat "$file1")

chars2=$(cat "$file2")

These lines read the contents of file1 and file2 into the variables chars1 and chars2, respectively. The cat command is used to concatenate and display the contents of files.

# Get the length of the longest string

len=$(( ${#chars1} > ${#chars2} ? ${#chars1} : ${#chars2} ))

This line compares the lengths of chars1 and chars2 using the ternary operator ? : and assigns the length of the longer string to the variable len. The ${#variable} syntax is used to get the length of a string in Bash.

# Combining the characters

combine=""

for ((i=0; i<len; i++)); do

if [ $i -lt ${#chars1} ]; then

combine+="${chars1:i:1}"

fi

if [ $i -lt ${#chars2} ]; then

combine+="${chars2:i:1}"

fi

Done

This block (for loop) of code combines the characters from chars1 and chars2 into a new string called combine. It iterates from 0 to len - 1 and appends characters from each string to combine. The ${chars1:i:1} syntax extracts a single character at index i from chars1.

# Write the combined characters to the output file

echo "$combine" > "$output\_file"

This line writes the contents of combine to the output\_file using the echo command and the output redirection operator (>). The > operator overwrites the contents of the file with the echoed string.

echo "combine characters have been written to '$output\_file'."

Finally, this line prints a message to the console, indicating that the combine characters have been written to the output\_file.

Result I get this is

