Assignment 7:

Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

Sol:

#!/bin/bash

Check if the correct number of arguments is provided

if ["$#" -ne 3]; then

   echo "Usage: $0 <input_file> <old_text> <new_text>"

   exit 1

fi

Input file

input_file="$1"

Old text to replace

old_text="$2"

New text to replace with

new_text="$3"

Check if input file exists

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

   echo "Error: Input file '$input_file' does not exist."

   exit 1

fi

Output file

output_file="${input_file%.txt}_replaced.txt"

Replace all occurrences of old_text with new_text using sed and write the result to the output file

sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"

echo "Replacement completed. Result saved in '$output_file'."

Explanation:

Argument Check:

- The script checks if the correct number of arguments (3) is provided. If not, it displays usage information and exits.

Input File Existence Check:

- It verifies whether the input file exists. If not, it displays an error message and exits.

Input Arguments:

- The script takes three arguments:
- `<input_file>:` The path to the input text file.
- `<old_text>:` The text to be replaced.
- `<new_text>:` The text to replace `<old_text>` with.

Output File Path:

- It constructs the output file path by appending "_replaced.txt" to the input file name.

Replacement with `sed`:

- It uses `sed` to perform the replacement operation.
- The `s` command in `sed` is used to substitute occurrences of a pattern (in this case, `$old_text`) with a replacement (in this case, `$new_text`). The `g` flag ensures that all occurrences are replaced, not just the first one.
- The result of the `sed` operation is then redirected to the output file.

Output Message:

- Finally, it prints a message indicating that the replacement is completed and specifies the path to the output file where the result is saved.