**Write a Menu-Driven Shell Script for String Manipulations Without Using Predefined Functions**

AIM : To create a **menu-driven shell script** that performs various string manipulations **without using predefined functions**

#### ****Step 1: Start****

* Begin the execution of the shell script.
* Display a welcome message and instructions for the user.

#### ****Step 2: Display the menu options****

* Show the list of string operations that the user can choose from:
  1. Convert to Uppercase
  2. Convert to Lowercase
  3. Find String Length
  4. Reverse a String
  5. Extract a Substring
  6. Replace a Word in a String
  7. Count Occurrences of a Character/Word
  8. Check if a String is Empty or Not
  9. Remove Whitespaces from a String
  10. Split a String by a Delimiter
  11. Append a Character to a String
  12. Check if Two Strings are Equal
  13. Exit

#### ****Step 3: Take User Input****

* Prompt the user to enter their choice.
* Store the choice in a variable.

#### ****Step 4: Perform Operations Based on User Choice****

##### ****Case 1: Convert to Uppercase****

* Read the string input from the user.
* Iterate over each character and check its ASCII value.
* If the character is lowercase (a-z), subtract **32** from its ASCII value to convert it to uppercase.
* Store the modified characters and display the new string.

##### ****Case 2: Convert to Lowercase****

* Read the string input from the user.
* Iterate over each character and check its ASCII value.
* If the character is uppercase (A-Z), add **32** to its ASCII value to convert it to lowercase.
* Store the modified characters and display the new string.

##### ****Case 3: Find String Length****

* Read the string input from the user.
* Initialize a counter variable (length=0).
* Use a loop to iterate through each character, incrementing the counter.
* Display the final count as the length of the string.

##### ****Case 4: Reverse a String****

* Read the string input from the user.
* Use a loop to iterate over the string **from the last character to the first** and print each character.

##### ****Case 5: Extract a Substring****

* Read the string from the user.
* Ask for starting position and length.
* Use a loop to extract characters **starting from the given index** and store them in a new string.
* Display the extracted substring.

##### ****Case 6: Replace a word in a String****

* Read the original string, the word to be replaced, and the new word.
* Use loops to check if the word exists by **matching character by character**.
* Replace it with the new word manually by **constructing a new string**.
* Display the modified string.

##### ****Case 7: Count Occurrences of a Character/Word****

* Read the string and the character/word to be counted.
* Use a loop to iterate over the string and compare **each character or word**.
* Keep a counter and increment it whenever a match is found.
* Display the total count.

##### ****Case 8: Check if a String is Empty****

* Read the string from the user.
* If the first character is **null**, print "String is empty."
* Otherwise, print "String is not empty."

##### ****Case 9: Remove Whitespaces from a String****

* Read the string from the user.
* Use a loop to iterate over the characters and **skip spaces** while forming a new string.
* Display the string without spaces.

##### ****Case 10: Split a String by a Delimiter****

* Read the string and the delimiter.
* Iterate over the string and split words wherever the delimiter is found.
* Store and display each part separately.

##### ****Case 11: Append a Character to a String****

* Read the string and the character to append.
* Manually concatenate by adding the character to the end of the string.
* Display the new string.

##### ****Case 12: Check if Two Strings are Equal****

* Read two strings from the user.
* Compare each character one by one using a loop.
* If all characters match, print "Strings are equal."
* Otherwise, print "Strings are not equal."

##### ****Case 13: Exit****

* Display a goodbye message.
* Terminate the script.

#### ****Step 5: Repeat Until Exit****

* Return to the menu and allow the user to select another operation.
* If the user selects exit, stop the program.

#### ****Step 6: Stop****

* End the execution of the script.

**Source Code:**

#!/bin/bash

while true; do

echo "===== STRING MANIPULATION MENU ====="

echo "1. Convert to Uppercase"

echo "2. Convert to Lowercase"

echo "3. Find String Length"

echo "4. Reverse a String"

echo "5. Extract a Substring"

echo "6. Replace a Word in a String"

echo "7. Count Occurrences of a Character/Word"

echo "8. Check if a String is Empty or Not"

echo "9. Remove Whitespaces from a String"

echo "10. Split a String by a Delimiter"

echo "11. Append a Character to a String"

echo "12. Check if Two Strings are Equal"

echo "13. Exit"

read -p "Enter your choice: " choice

case $choice in

1. read -p "Enter a string: " str

result=""

for ((i=0; i<${#str}; i++)); do

char="${str:$i:1}"

case "$char" in

[a-z]) result+=$(echo "$char" | awk '{printf("%c", toupper($1))}') ;;

\*) result+="$char" ;;

esac

done

echo "Uppercase: $result" ;;

1. read -p "Enter a string: " str

result=""

for ((i=0; i<${#str}; i++)); do

char="${str:$i:1}"

case "$char" in

[A-Z]) result+=$(echo "$char" | awk '{printf("%c", tolower($1))}') ;;

\*) result+="$char" ;;

esac

done

echo "Lowercase: $result" ;;

1. read -p "Enter a string: " str

length=0

while [ -n "$str" ]; do

str=${str#?}

((length++))

done

echo "String Length: $length" ;;

1. read -p "Enter a string: " str

reversed=""

for ((i=${#str}-1; i>=0; i--)); do

reversed+="${str:$i:1}"

done

echo "Reversed String: $reversed" ;;

1. read -p "Enter a string: " str

read -p "Enter starting position: " pos

read -p "Enter length: " len

substr=""

for ((i=pos; i<pos+len && i<${#str}; i++)); do

substr+="${str:$i:1}"

done

echo "Substring: $substr" ;;

1. read -p "Enter the original string: " str

read -p "Enter the word to replace: " old

read -p "Enter the new word: " new

result=""

words=($str)

for word in "${words[@]}"; do

if [ "$word" == "$old" ]; then

result+="$new "

else

result+="$word "

fi

done

echo "Updated String: $result" ;;

1. read -p "Enter a string: " str

read -p "Enter the character/word to count: " search

count=0

words=($str)

for word in "${words[@]}"; do

if [ "$word" == "$search" ]; then

((count++))

fi

done

echo "Occurrences: $count" ;;

1. read -p "Enter a string: " str

if [ -z "$str" ]; then

echo "The string is empty."

else

echo "The string is not empty."

fi ;;

1. read -p "Enter a string: " str

trimmed=""

for ((i=0; i<${#str}; i++)); do

char="${str:$i:1}"

if [ "$char" != " " ]; then

trimmed+="$char"

fi

done

echo "String without whitespaces: $trimmed" ;;

1. read -p "Enter a string: " str

read -p "Enter delimiter: " delimiter

temp=""

for ((i=0; i<${#str}; i++)); do

char="${str:$i:1}"

if [ "$char" == "$delimiter" ]; then

echo "$temp"

temp=""

else

temp+="$char"

fi

done

echo "$temp" ;;

1. read -p "Enter a string: " str

read -p "Enter character to append: " char

echo "Updated String: ${str}${char}" ;;

1. read -p "Enter first string: " str1

read -p "Enter second string: " str2

if [ "$str1" == "$str2" ]; then

echo "Strings are equal."

else

echo "Strings are not equal."

fi ;;

1. echo "Exiting..."

exit 0 ;;

\*) echo "Invalid choice. Please try again." ;;

esac

echo "--------------------------------------"

done

**Write a Menu-Driven Shell Script for String Manipulations Using Predefined Functions**

### ****AIM****

To create a **menu-driven shell script** that performs various string manipulations using predefined functions.

**Algorithm**

#### ****Step 1: Start****

* Begin the execution of the shell script.
* Display a welcome message and instructions for the user.

#### ****Step 2: Display the menu options****

* Show the list of string operations that the user can choose from:
  1. Convert to Uppercase
  2. Convert to Lowercase
  3. Find String Length
  4. Reverse a String
  5. Extract a Substring
  6. Replace a Word in a String
  7. Count Occurrences of a Character/Word
  8. Check if a String is Empty or Not
  9. Remove Whitespaces from a String
  10. Split a String by a Delimiter
  11. Append a Character to a String
  12. Check if Two Strings are Equal
  13. Exit

#### ****Step 3: Take User Input****

* Prompt the user to enter their choice.
* Store the choice in a variable.

#### ****Step 4: Perform Operations Based on User Choice****

##### ****Case 1: Convert to Uppercase****

* Read the string input from the user.
* Use the tr command to convert all lowercase letters (a-z) to uppercase (A-Z).
* Display the uppercase string.
  + **Command:** echo "$str" | tr '[:lower:]' '[:upper:]'

##### ****Case 2: Convert to Lowercase****

* Read the string input from the user.
* Use the tr command to convert all uppercase letters (A-Z) to lowercase (a-z).
* Display the lowercase string.
  + **Command:** echo "$str" | tr '[:upper:]' '[:lower:]'

##### ****Case 3: Find String Length****

* Read the string input from the user.
* Use ${#str} to calculate the length of the string.
* Display the string length.
  + **Command:** echo ${#str}

##### ****Case 4: Reverse a String****

* Read the string input from the user.
* Use the rev command to reverse the string.
* Display the reversed string.
  + **Command:** echo "$str" | rev

##### ****Case 5: Extract a Substring****

* Read the string from the user.
* Ask for the starting position and the length.
* Use ${str:pos:len} to extract the substring.
* Display the extracted substring.
  + **Command:** echo ${str:pos:len}

##### ****Case 6: Replace a Word in a String****

* Read the original string, the word to be replaced, and the new word.
* Use the sed command to replace occurrences of the word.
* Display the modified string.
  + **Command:** echo "$str" | sed "s/$old/$new/g"

##### ****Case 7: Count Occurrences of a Character/Word****

* Read the string and the character/word to be counted.
* Use the grep -o command to count occurrences.
* Display the total count.
  + **Command:** echo "$str" | grep -o "$search" | wc -l

##### ****Case 8: Check if a String is Empty****

* Read the string from the user.
* Use [ -z "$str" ] to check if the string is empty.
* Display the appropriate message.

if [ -z "$str" ]; then

echo "The string is empty."else

echo "The string is not empty."fi

##### ****Case 9: Remove Whitespaces from a String****

* Read the string from the user.
* Use tr -d '[:space:]' to remove all whitespace characters.
* Display the modified string.
  + **Command:** echo "$str" | tr -d '[:space:]'

##### ****Case 10: Split a String by a Delimiter****

* Read the string and the delimiter from the user.
* Use IFS (Internal Field Separator) to split the string.
* Display each part separately.

IFS="$delimiter" read -ra parts <<< "$str"for part in "${parts[@]}"; do

echo "$part"done

##### ****Case 11: Append a Character to a String****

* Read the string and the character to append.
* Use direct concatenation to append the character to the string.
* Display the updated string.
  + **Command:** echo "${str}${char}"

##### ****Case 12: Check if Two Strings are Equal****

* Read two strings from the user.
* Use [ "$str1" == "$str2" ] to compare the strings.
* Display whether they are equal or not.

**Command**

if [ "$str1" == "$str2" ]; then

echo "Strings are equal."else

echo "Strings are not equal."fi

##### ****Case 13: Exit****

* Display a goodbye message.
* Terminate the script using exit 0.

#### ****Step 5: Repeat Until Exit****

* Return to the menu and allow the user to select another operation.
* If the user selects exit, stop the program.

#### ****Step 6: Stop****

* End the execution of the script.

**Source Code**

#!/bin/bash

while true; do

echo "===== STRING MANIPULATION MENU ====="

echo "1. Convert to Uppercase"

echo "2. Convert to Lowercase"

echo "3. Find String Length"

echo "4. Reverse a String"

echo "5. Extract a Substring"

echo "6. Replace a Word in a String"

echo "7. Count Occurrences of a Character/Word"

echo "8. Check if a String is Empty or Not"

echo "9. Remove Whitespaces from a String"

echo "10. Split a String by a Delimiter"

echo "11. Append a Character to a String"

echo "12. Check if Two Strings are Equal"

echo "13. Exit"

read -p "Enter your choice: " choice

case $choice in

1. read -p "Enter a string: " str

echo "Uppercase: $(echo "$str" | tr '[:lower:]' '[:upper:]')" ;;

1. read -p "Enter a string: " str

echo "Lowercase: $(echo "$str" | tr '[:upper:]' '[:lower:]')” ;;

1. read -p "Enter a string: " str

echo "String Length: ${#str}" ;;

1. read -p "Enter a string: " str

echo "Reversed String: $(echo "$str" | rev)" ;;

1. read -p "Enter a string: " str

read -p "Enter starting position: " pos

read -p "Enter length: " len

echo "Substring: ${str:pos:len}" ;;

1. read -p "Enter the original string: " str

read -p "Enter the word to replace: " old

read -p "Enter the new word: " new

echo "Updated String: $(echo "$str" | sed "s/$old/$new/g")" ;;

1. read -p "Enter a string: " str

read -p "Enter the character/word to count: " search

count=$(echo "$str" | grep -o "$search" | wc -l)

echo "Occurrences: $count" ;;

1. read -p "Enter a string: " str

if [ -z "$str" ]; then

echo "The string is empty."

else

echo "The string is not empty."

fi ;;

1. read -p "Enter a string: " str

echo "String without whitespaces: $(echo "$str" | tr -d '[:space:]')" ;;

1. read -p "Enter a string: " str

read -p "Enter delimiter: " delimiter

IFS="$delimiter" read -ra parts <<< "$str"

echo "Splitted String:"

for part in "${parts[@]}"; do

echo "$part"

done ;;

1. read -p "Enter a string: " str

read -p "Enter character to append: " char

echo "Updated String: ${str}${char}" ;;

12) read -p "Enter first string: " str1

read -p "Enter second string: " str2

if [ "$str1" == "$str2" ]; then

echo "Strings are equal."

else

echo "Strings are not equal."

fi ;;

1. echo "Exiting..."

exit 0 ;;

\*) echo "Invalid choice. Please try again." ;;

esac

echo "--------------------------------------"

done