**1. Reverse a String**

#include <stdio.h>

#include <string.h>

void reverseString(char str[]) {

int start = 0;

int end = strlen(str) - 1;

while(start < end) {

char temp = str[start];

str[start] = str[end];

str[end] = temp;

start++;

end--;

}

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin); // To handle spaces in the string

str[strcspn(str, "\n")] = '\0'; // Remove trailing newline

reverseString(str);

printf("Reversed string: %s\n", str);

return 0;

}

### ****2. Check if a String is a Palindrome****

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int isPalindrome(char str[]) {

int start = 0;

int end = strlen(str) - 1;

while (start < end) {

if (tolower(str[start]) != tolower(str[end])) {

return 0; // Not a palindrome

}

start++;

end--;

}

return 1; // Palindrome

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

if (isPalindrome(str)) {

printf("Yes, it is a palindrome.\n");

} else {

printf("No, it is not a palindrome.\n");

}

return 0;

}

### ****3. Count Vowels and Consonants****

#include <stdio.h>

#include <ctype.h>

void countVowelsAndConsonants(char str[]) {

int vowels = 0, consonants = 0;

for(int i = 0; str[i] != '\0'; i++) {

if (isalpha(str[i])) {

char ch = tolower(str[i]);

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowels++;

} else {

consonants++;

}

}

}

printf("Vowels: %d\n", vowels);

printf("Consonants: %d\n", consonants);

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

countVowelsAndConsonants(str);

return 0;

}

### ****4. Find the Length of a String****

#include <stdio.h>

int stringLength(char str[]) {

int length = 0;

while (str[length] != '\0') {

length++;

}

return length;

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

printf("Length: %d\n", stringLength(str));

return 0;

}

### ****5. Remove All Spaces from a String****

#include <stdio.h>

void removeSpaces(char str[]) {

int i = 0, j = 0;

while (str[i]) {

if (str[i] != ' ') {

str[j++] = str[i];

}

i++;

}

str[j] = '\0';

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

removeSpaces(str);

printf("String without spaces: %s\n", str);

return 0;

}

### ****6. Find the First Non-Repeated Character****

#include <stdio.h>

#include <string.h>

char firstNonRepeatedChar(char str[]) {

int count[256] = {0}; // ASCII size, assuming ASCII characters

for (int i = 0; str[i] != '\0'; i++) {

count[(int)str[i]]++;

}

for (int i = 0; str[i] != '\0'; i++) {

if (count[(int)str[i]] == 1) {

return str[i];

}

}

return '\0'; // No non-repeated character

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

char result = firstNonRepeatedChar(str);

if (result != '\0') {

printf("First non-repeated character: %c\n", result);

} else {

printf("No non-repeated character.\n");

}

return 0;

}

### ****7. Count the Number of Words in a String****

#include <stdio.h>

#include <ctype.h>

int countWords(char str[]) {

int count = 0, inWord = 0;

for (int i = 0; str[i] != '\0'; i++) {

if (isspace(str[i]) || str[i] == '\0') {

inWord = 0;

} else if (inWord == 0) {

count++;

inWord = 1;

}

}

return count;

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

printf("Word count: %d\n", countWords(str));

return 0;

}

### ****8. Convert a String to Uppercase and Lowercase****

#include <stdio.h>

#include <ctype.h>

void convertCase(char str[]) {

for (int i = 0; str[i] != '\0'; i++) {

if (isupper(str[i])) {

str[i] = tolower(str[i]);

} else if (islower(str[i])) {

str[i] = toupper(str[i]);

}

}

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0';

convertCase(str);

printf("Converted string: %s\n", str);

return 0;

}

### ****1. Manual Reversal of a String****

#include <stdio.h>

#include <string.h>

void reverseString(char str[]) {

int start = 0;

int end = strlen(str) - 1;

// Loop to swap characters from both ends

while (start < end) {

char temp = str[start];

str[start] = str[end];

str[end] = temp;

start++;

end--;

}

}

int main() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

str[strcspn(str, "\n")] = '\0'; // Remove trailing newline from input

reverseString(str);

printf("Reversed string: %s\n", str);

return 0;

}

### ****How to remove char in String****

#include <stdio.h>

#include <string.h>

void removeChar(char str[], char ch) {

int i, j = 0;

// Traverse the string

for (i = 0; str[i] != '\0'; i++) {

// If the current character is not the one to be removed, copy it

if (str[i] != ch) {

str[j++] = str[i];

}

}

// Null-terminate the modified string

str[j] = '\0';

}

int main() {

char str[100], ch;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin); // Read the string

str[strcspn(str, "\n")] = '\0'; // Remove newline from input if present

printf("Enter the character to remove: ");

scanf("%c", &ch); // Read the character to be removed

removeChar(str, ch);

printf("Modified string: %s\n", str);

return 0;

}