/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* DAY 14: STRING HANDLING – PART 1 \*

\* Contains 10 C programs for basic string operations \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <ctype.h> // For toupper(), tolower()

// ==============================

// 1. Read and Display a String

// ==============================

void readDisplayString() {

char str[100];

printf("Enter a string: ");

fgets(str, sizeof(str), stdin); // Safer than gets()

printf("You entered: %s", str);

}

// ==============================

// 2. String Length Without strlen()

// ==============================

void stringLength() {

char str[100];

int length = 0;

printf("Enter a string: ");

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

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

length++;

}

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

}

// ==============================

// 3. Reverse a String

// ==============================

void reverseString() {

char str[100], temp;

int start = 0, end;

printf("Enter a string: ");

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

// Calculate length

end = 0;

while (str[end] != '\0' && str[end] != '\n') end++;

end--;

// Reverse logic

while (start < end) {

temp = str[start];

str[start] = str[end];

str[end] = temp;

start++;

end--;

}

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

}

// ==============================

// 4. Convert to Uppercase

// ==============================

void stringToUpper() {

char str[100];

int i = 0;

printf("Enter a string: ");

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

while (str[i]) {

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

i++;

}

printf("Uppercase: %s", str);

}

// ==============================

// 5. Convert to Lowercase

// ==============================

void stringToLower() {

char str[100];

int i = 0;

printf("Enter a string: ");

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

while (str[i]) {

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

i++;

}

printf("Lowercase: %s", str);

}

// ==============================

// 6. Check String Palindrome

// ==============================

void checkStringPalindrome() {

char str[100];

int start = 0, end, isPalindrome = 1;

printf("Enter a string: ");

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

end = 0;

while (str[end] != '\0' && str[end] != '\n') end++;

end--;

while (start < end) {

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

isPalindrome = 0;

break;

}

start++;

end--;

}

printf("%s is %s\n", str, isPalindrome ? "Palindrome" : "Not Palindrome");

}

// ==============================

// 7. Count Vowels and Consonants

// ==============================

void countVowelsConsonants() {

char str[100];

int vowels = 0, consonants = 0, i = 0;

printf("Enter a string: ");

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

while (str[i]) {

char ch = tolower(str[i]);

if (ch >= 'a' && ch <= 'z') {

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

vowels++;

else

consonants++;

}

i++;

}

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

}

// ==============================

// 8. Count Words in String

// ==============================

void countWords() {

char str[100];

int words = 0, i = 0;

printf("Enter a string: ");

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

while (str[i]) {

if (str[i] == ' ' || str[i] == '\n' || str[i] == '\t') {

words++;

}

i++;

}

printf("Words: %d\n", words);

}

// ==============================

// 9. Character Frequency

// ==============================

void charFrequency() {

char str[100], ch;

int count = 0, i = 0;

printf("Enter a string: ");

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

printf("Enter character to find: ");

scanf("%c", &ch);

while (str[i]) {

if (str[i] == ch) {

count++;

}

i++;

}

printf("'%c' appears %d times\n", ch, count);

}

// ==============================

// 10. String Copy Without strcpy()

// ==============================

void stringCopy() {

char src[100], dest[100];

int i = 0;

printf("Enter source string: ");

fgets(src, sizeof(src), stdin);

while (src[i] != '\0') {

dest[i] = src[i];

i++;

}

dest[i] = '\0';

printf("Copied string: %s", dest);

}