# ✅ Day : Strings (7-8-2025)

1. Write a program to find the length of a string without using strlen().

Input : A string entered by the user

Process : 1. Initialize a counter to 0.  
2. Traverse each character of the string until the null character '\0' is found.  
3. Increment counter for each character

Output : Length of the string

Program :

#include <stdio.h>

void main()

{

char str[100];

int i = 0;

scanf("%[^\n]", str);

while (str[i] != '\0')

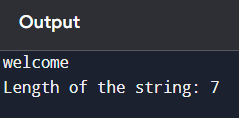
{

i++;

}

printf("Length of the string: %d\n", i);

}



1. Write a program to copy one string to another.

**Input:**

* A string from the user.

**Process:**

1. Read the string into str1.
2. Copy each character from str1 to str2 until '\0'.
3. Add '\0' at the end of str2.

**Output:**

* The copied string.

Program :

#include<stdio.h>

void main()

{

char str1[10]="welcome";

char str2[10];

int i,l=6;

for(i=0;i<10;i++)

{

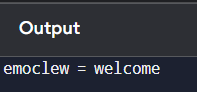
str2[i]=str1[l];

l--;

}

printf("%s",str2);

}



1. Write a program to concatenate two strings.

**Input:**

* Two strings from the user.

**Process:**

1. Read first string into str1 and second string into str2.
2. Move to the end of str1.
3. Copy characters from str2 to the end of str1 until '\0'.
4. Add '\0' at the end.

**Output:**

* The concatenated string.

Program :

#include<stdio.h>

void main()

{

char str1[10]="welcome";

char str2[10]="india";

char str3[10];

int i;

for(i=0;i<7;i++)

{

str3[i]=str1[i];

}

for(i=0;i<7;i++)

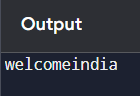
{

str3[i+7]=str2[i];

}

printf("%s",str3);

}



1. Write a program to compare two strings.

* Two strings from the user.

**Process:**

1. Read both strings into str1 and str2.
2. Compare characters one by one.
3. If all characters match and lengths are same → strings are equal.
4. Otherwise → strings are not equal.

**Output:**

* Message showing whether strings are equal or not.

Program :

#include <stdio.h>

int main()

{

char str1[100], str2[100];

int i = 0, flag = 0;

printf("Enter first string: ");

scanf("%[^\n]", str1);

getchar();

printf("Enter second string: ");

scanf("%[^\n]", str2);

while (str1[i] != '\0' || str2[i] != '\0')

{

if (str1[i] != str2[i]) {

flag = 1;

break;

}

i++;

}

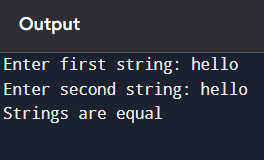
if (flag == 0)

printf("Strings are equal\n");

else

printf("Strings are not equal\n");

}



1. Write a program to count vowels and consonants in a string

**input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Initialize vowelCount = 0, consonantCount = 0.
3. Traverse each character:
   * If it is a vowel (a, e, i, o, u in upper or lower case) → increment vowelCount.
   * If it is an alphabet but not a vowel → increment consonantCount.
4. Ignore non-alphabet characters.

**Output:**

* Number of vowels and consonants.

Program :

#include <stdio.h>

#include <ctype.h>

void main() {

char str[100];

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

char ch;

printf("Enter a string: ");

scanf("%[^\n]", str);

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

ch = tolower(str[i]);

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

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

vowels++;

else

consonants++;

}

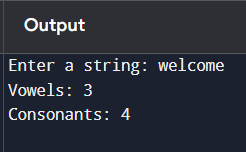
i++;

}

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

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

}



.

1. Write a program to convert lowercase to uppercase and vice versa.

**Input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Traverse each character:
   * If lowercase → convert to uppercase.
   * If uppercase → convert to lowercase.
   * Otherwise → leave unchanged.

**Output:**

* Modified string with cases swapped.

Program :

#include <stdio.h>

#include <ctype.h>

void main()

{

char str[100];

int i = 0;

printf("Enter a string: ");

scanf("%[^\n]", str);

while (str[i] != '\0')

{

if (islower(str[i]))

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

else if (isupper(str[i]))

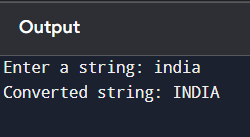
str[i] = tolower(str[i]);

i++;

}

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

}



1. Write a program to check if a string is palindrome

**Input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Initialize two indexes: start at 0, end at length-1.
3. Compare characters from start and end moving toward the middle.
4. If all match → string is palindrome, else not.

**Output:**

* Message stating whether the string is palindrome or not.

Program :

#include <stdio.h>

#include <string.h>

void main()

{

char str[100];

int start, end, flag = 0;

printf("Enter a string: ");

scanf("%[^\n]", str);

start = 0;

end = strlen(str) - 1;

while (start < end)

{

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

{

flag = 1;

}

start++;

end--;

}

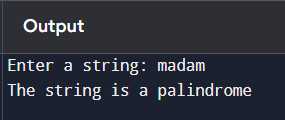
if (flag == 0)

printf("The string is a palindrome\n");

else

printf("The string is not a palindrome\n");

}



1. Write a program to reverse a string.

**Input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Find the length of the string.
3. Swap the first and last characters, moving toward the center.
4. Continue until the string is reversed.

**Output:**

* The reversed string.

Program :

#include <stdio.h>

int main()

{

char str[100], temp;

int start = 0, end = 0;

printf("Enter a string: ");

scanf("%[^\n]", str);

while (str[end] != '\0')

{

end++;

}

end--;

while (start < end)

{

temp = str[start];

str[start] = str[end];

str[end] = temp;

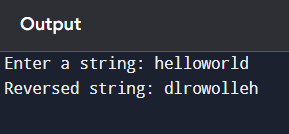
start++;

end--;

}

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

}



1. Write a program to count words in a string.

**Input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Initialize count = 0.
3. Traverse the string character by character.
4. If a space is found followed by a non-space → increment count.
5. Add 1 to count for the first word (if string is not empty).

**Output:**

* Number of words in the string.

Program :

#include <stdio.h>

void main()

{

char str[200];

int i = 0, count = 0;

printf("Enter a string: ");

scanf("%[^\n]", str);

if (str[0] != ' ' && str[0] != '\0')

count = 1;

while (str[i] != '\0')

{

if (str[i] == ' ' && str[i + 1] != ' ' && str[i + 1] != '\0')

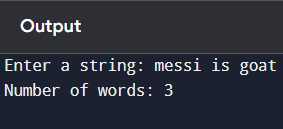
count++;

i++;

}

printf("Number of words: %d\n", count);

}



Write a program to find the frequency of each character in a string.

**Input:**

* A string from the user.

**Process:**

1. Read the string into str.
2. Create an integer array freq[256] initialized to 0 (for all ASCII characters).
3. Traverse each character of str and increment its frequency in freq.
4. Traverse freq and print characters with non-zero counts.

**Output:**

* Each character in the string with its frequency.

Program :

#include <stdio.h>

int main()

{

char str[200];

int freq[256] = {0}; // ASCII size

int i;

printf("Enter a string: ");

scanf("%[^\n]", str);

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

{

freq[(unsigned char)str[i]]++;

}

printf("Character frequencies:\n");

for (i = 0; i < 256; i++)

{

if (freq[i] != 0)

{

printf("%c : %d\n", i, freq[i]);

}

}

}

