DAY 7-8-2025

STRINGS

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

 Input: A string from user

 Process: Traverse the string until \0, count characters

 Output: Display the length

CODE:

#include <stdio.h>

void main()

{

char str[100];

int i = 0;

printf("Enter a string: ");

gets(str);

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

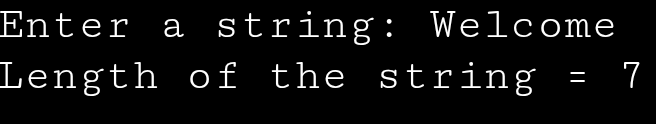
i++;

}

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

}

Output:



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

 Input: A source string

 Process: Copy each character to destination

 Output: Display copied string

CODE:

#include <stdio.h>

void main()

{

char str1[100], str2[100];

int i;

printf("Enter a string: ");

gets(str1);

for (i = 0; str1[i] != '\0'; i++) {

str2[i] = str1[i];

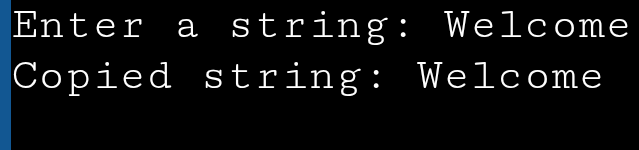
}

str2[i] = '\0';

printf("Copied string: %s\n", str2);

}

Output:



1. Write a program to concatenate two strings.

 Input: Two strings

 Process: Append second string to first

 Output: Concatenated string

CODE:

#include <stdio.h>

void main()

{

char str1[100], str2[100];

int i, j;

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

for (i = 0; str1[i] != '\0'; i++);

for (j = 0; str2[j] != '\0'; j++, i++) {

str1[i] = str2[j];

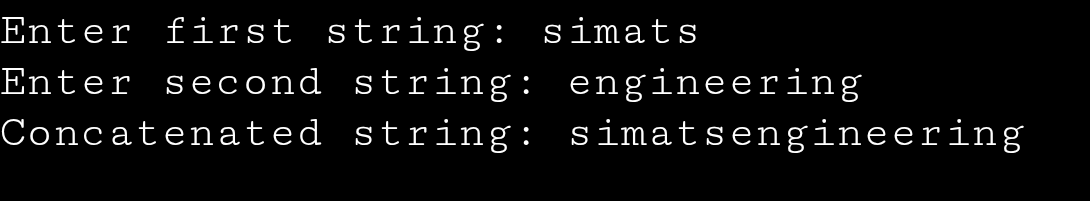
}

str1[i] = '\0';

printf("Concatenated string: %s\n", str1);

}

Output:



1. Write a program to compare two strings.

 Input: Two strings

 Process: Compare characters one by one

 Output: Show whether they are equal or not

CODE:

#include <stdio.h>

void main()

{

char str1[100], str2[100];

int i = 0, flag = 0;

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

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

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

flag = 1;

break;

}

i++;

}

if (flag == 0)

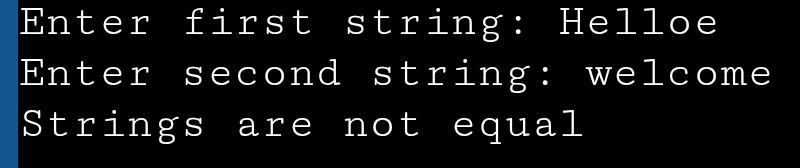
printf("Strings are equal\n");

else

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

}

Output:



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

 Input: A string

 Process: Check each character, count vowels/consonants

 Output: Display count

CODE:

#include <stdio.h>

void main()

{

char str[100];

int i, vowels = 0, consonants = 0;

printf("Enter a string: ");

gets(str);

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

char ch = str[i];

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

ch = (ch >= 'A' && ch <= 'Z') ? ch + 32 : ch;

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

vowels++;

else

consonants++;

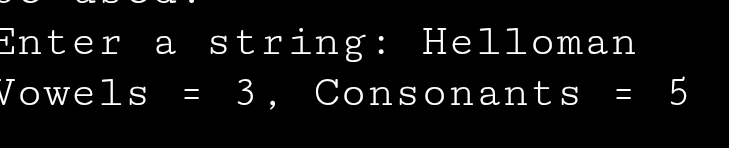
}

}

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

}

Output:



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

 Input: A string

 Process: Convert using ASCII

 Output: Converted string

CODE:

#include <stdio.h>

void main()

{

char str[100];

int i;

printf("Enter a string: ");

gets(str);

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

if (str[i] >= 'a' && str[i] <= 'z')

str[i] -= 32;

else if (str[i] >= 'A' && str[i] <= 'Z')

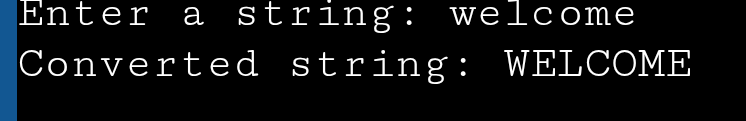
str[i] += 32;

}

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

}

Output:



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

 Input: A string

 Process: Reverse and compare

 Output: Palindrome or not

CODE:

#include <stdio.h>

#include <string.h>

int main() {

char str[100], rev[100];

int i, len, flag = 0;

printf("Enter a string: ");

gets(str);

len = strlen(str);

for (i = 0; i < len; i++) {

rev[i] = str[len - i - 1];

}

rev[i] = '\0';

if (strcmp(str, rev) == 0)

printf("Palindrome\n");

else

printf("Not a palindrome\n");

return 0;

}

Output:

A black background with white text

AI-generated content may be incorrect.

1. Write a program to reverse a string.

 Input: A string

 Process: Swap characters from both ends

 Output: Reversed string

CODE:

#include <stdio.h>

#include <string.h>

void main()

{

char str[100], temp;

int i, len;

printf("Enter a string: ");

gets(str);

len = strlen(str);

for (i = 0; i < len / 2; i++) {

temp = str[i];

str[i] = str[len - i - 1];

str[len - i - 1] = temp;

}

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

}

Output:



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

 Input: A string

 Process: Count spaces between words

 Output: Word count

CODE:

#include <stdio.h>

void main()

{

char str[100];

int i, count = 1;

printf("Enter a string: ");

gets(str);

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

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

count++;

}

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

}

Output:

A black background with white letters

AI-generated content may be incorrect.

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

 Input: A string

 Process: Use an array to count ASCII frequencies

 Output: Show character frequencies

CODE:

#include <stdio.h>

void main()

{

char str[100];

int freq[256] = {0}, i;

printf("Enter a string: ");

gets(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]);

}

}

Output:

