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

IPO

| **Component** | **Description** |
| --- | --- |
| **I (Input)** | A string (e.g., "welcome") — stored in a character array (char a[10]). |
| **P (Process)** | Loop through the string until the null character '\0' is found, counting each character. |
| **O (Output)** | The number of characters in the string (i.e., the **length** of the string). |

CODE

#include<stdio.h>

void main()

{

char a[10]="welcome";

int i,count=0;

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

{

if(a[i]!='\0')

count++;

}

printf("NUMBER OF LETTERS=%d",count);

}

OUTPUT



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

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A string (e.g., "welcome") stored in str[] |
| **Process** | Copy each character of str into str2 including \0 |
| **Output** | Printed copied string str2 |

CODE

#include<stdio.h>

void main()

{

char str[10]="welcome";

char str2[10];

int i,count=0;

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

{

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

count++;

}

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

{

str2[i]=str[i];

}

printf("COPY STRING=%s",str2);

}

OUTPUT



1. Write a program to concatenate two strings.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | Two strings: str = "welcome" and str2 = "home" |
| **Process** | Find the end of str, append each character of str2 to it, add '\0' at end |
| **Output** | Concatenated string: "welcomehome" |

CODE

#include<stdio.h>

void main()

{

char str[10]="welcome";

char str2[10]="home";

int i,j;

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

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

{

str[i+j]=str2[j];

}

str[i+j]='\0';

printf("%s",str);

}

OUTPUT



1. Write a program to compare two strings.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | Two strings from the user |
| **Process** | Compare character by character until mismatch or end of strings |
| **Output** | Message indicating whether the strings are equal or not |

CODE

#include <stdio.h>

void main()

{

char str1[50], str2[50];

int i = 0, flag = 0;

printf("Enter first string: ");

gets(str1); // or use fgets(str1, 50, stdin);

printf("Enter second string: ");

gets(str2); // or use fgets(str2, 50, stdin);

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

{

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

{

flag = 1; // strings are not equal

break;

}

i++;

}

if(str1[i] != '\0' || str2[i] != '\0') // different lengths

flag = 1;

if(flag == 0)

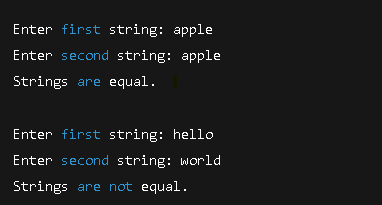
printf("Strings are equal.\n");

else

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

}

OUTPUT



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

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A predefined string: "welcome" |
| **Process** | Loop through string, check each character: vowels → vcount++, else → ccount++ |
| **Output** | Print number of vowels and consonants |

CODE

]

#include<stdio.h>

void main()

{

char str[10]="welcome";

int i,vcount=0,ccount=0;

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

{

if(str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u')

vcount++;

else

ccount++;

}

printf("VOWEL=%d\nCONSONANT%d\n",vcount,ccount);

}

OUPUT



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

IPO

| **Input** | **A string from the user (e.g., "WElcome123")** |
| --- | --- |
| **Process** | For each character: Upper → Lower Lower → Upper |
| **Output** | String with reversed letter cases |

CODE

#include<stdio.h>

void main()

{

char str[10]="WElcome";

int i,a;

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

{

a=str[i];

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

{

str[i]=str[i]+32;

}

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

{

str[i]=str[i]-32;

}

}

printf("%s",str);

}

OUTPUT



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

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A predefined string: "hannah" (stored in str[]) |
| **Process** | 1. Count the number of characters in str (excluding \0) 2. Reverse the string into str2 3. Compare both strings character by character 4. If all characters match, it's a palindrome |
| **Output** | Print the reversed string and whether it is a palindrome |

CODE

#include<stdio.h>

void main()

{

char str[10]="hannah";

char str2[10];

int i,count=0;

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

{

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

count++;

}

printf("%d\n",count);

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

{

str2[i]=str[count-1-i];

}

str2[i]='\0';

printf("%s\n",str2);

int flag=0;

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

{

if(str[i]!=str2[i])

{

flag=1;

break;

}

}

if(flag==0)

{

printf("its a palindrome");

}

else

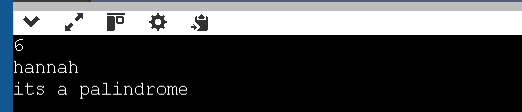
{

printf("its not a palindrome");

}

}

OUTPUT



1. Write a program to reverse a string.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A predefined string: "welcome" (stored in str[]) |
| **Process** | 1. Count the characters in str (excluding '\0')  2. Reverse the string into str2 |
| **Output** | Prints:  – Length of the string – Reversed string |

CODE

#include<stdio.h>

void main()

{

char str[10]="welcome";

char str2[10];

int i,count=0;

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

{

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

count++;

}

printf("%d\n",count);

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

{

str2[i]=str[count-1-i];

}

str2[i]='\0';

printf("%s\n",str2);

}

OUTPUT



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

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A predefined string: "welcome to sse" |
| **Process** | Count the number of spaces (' ') in the string, then add 1 to get word count |
| **Output** | Number of words in the string |

CODE

#include<stdio.h>

void main()

{

char str[14]="welcome to sse";

int i,count=0;

for(i=0;i<14;i++)

{

if(str[i]==' ')

count++;

}

printf("%d\n",count+1);

}

OUTPUT



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

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | n (size of array), array elements |
| **Process** | For each unique element, count how many times it appears |
| **Output** | Print each unique element with its frequency |

CODE

#include<stdio.h>

void main()

{

int i,j,count=0,n,a[10];

scanf("%d",&n);

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

{

scanf("%d",&a[i]);

}

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

{

count=1;

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

{

if(a[i]==a[j])

break;

}

if(j<i)

continue;

for(j=1+i;j<n;j++)

{

if(a[i]==a[j])

count++;

}

printf("Occurence=%d\nTimes=%d\n",a[i],count);

}

}

OUTPUT

