Assignment(7/8/2025)

### ✅ 1. Find length of a string without using strlen()

**IPO:**

* Input: A string
* Process: Count characters until null \0
* Output: Length of string

**C Program:**

#include <stdio.h>

int main() {

char str[100];

int i = 0;

printf("Enter a string: ");

gets(str);

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

i++;

}

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

return 0;

}

**Sample Output:**

Enter a string: Hello

Length of string = 5

### ✅ 2. Copy one string to another

**IPO:**

* Input: A string
* Process: Copy character-by-character
* Output: Copied string

**C Program:**

#include <stdio.h>

int 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);

return 0;

}

**Sample Output:**

Enter a string: Welcome

Copied string: Welcome

### ✅ 3. Concatenate two strings

**IPO:**

* Input: Two strings
* Process: Append second string to first
* Output: Concatenated string

**C Program:**

#include <stdio.h>

int main() {

char str1[100], str2[50];

int i = 0, j = 0;

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

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

while (str2[j] != '\0') {

str1[i++] = str2[j++];

}

str1[i] = '\0';

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

return 0;

}

**Sample Output:**

Enter first string: Hello

Enter second string: World

Concatenated string: HelloWorld

### ✅ 4. Compare two strings

**IPO:**

* Input: Two strings
* Process: Compare character-by-character
* Output: Same or Different

**C Program:**

#include <stdio.h>

int 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");

return 0;

}

**Sample Output:**

Enter first string: apple

Enter second string: apple

Strings are equal

### ✅ 5. Count vowels and consonants in a string

**IPO:**

* Input: A string
* Process: Count vowels (a, e, i, o, u) and consonants
* Output: Number of vowels and consonants

**C Program:**

#include <stdio.h>

int 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 = tolower(ch);

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

vowels++;

else

consonants++;

}

}

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

return 0;

}

**Sample Output:**

Enter a string: Hello World

Vowels = 3, Consonants = 7

### ✅ 6. Convert lowercase to uppercase and vice versa

**IPO:**

* Input: A string
* Process: Convert each lowercase to uppercase and vice versa
* Output: Converted string

**C Program:**

#include <stdio.h>

int 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] = str[i] - 32;

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

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

}

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

return 0;

}

**Sample Output:**

Enter a string: Hello123

Converted string: hELLO123

### ✅ 7. Check if a string is palindrome

**IPO:**

* Input: A string
* Process: Compare string with its reverse
* Output: Palindrome or not

**C Program:**

#include <stdio.h>

#include <string.h>

int main() {

char str[100], rev[100];

int i, len, flag = 1;

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 Palindrome\n");

return 0;

}

**Sample Output:**

Enter a string: madam

Palindrome

### ✅ 8. Reverse a string

**IPO:**

* Input: A string
* Process: Reverse the string using index
* Output: Reversed string

**C Program:**

#include <stdio.h>

#include <string.h>

int main() {

char str[100], rev[100];

int i, len;

printf("Enter a string: ");

gets(str);

len = strlen(str);

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

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

}

rev[i] = '\0';

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

return 0;

}

**Sample Output:**

Enter a string: world

Reversed string: dlrow

### ✅ 9. Count words in a string

**IPO:**

* Input: A string
* Process: Count spaces between words
* Output: Word count

**C Program:**

#include <stdio.h>

int main() {

char str[100];

int i, words = 1;

printf("Enter a string: ");

gets(str);

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

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

words++;

}

printf("Total words = %d\n", words);

return 0;

}

**Sample Output:**

Enter a string: I love coding

Total words = 3

### ✅ 10. Frequency of each character in a string

**IPO:**

* Input: A string
* Process: Count how many times each character appears
* Output: Frequency of each character

**C Program:**

#include <stdio.h>

#include <string.h>

int 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]);

}

return 0;

}

**Sample Output:**

Enter a string: apple

Character Frequencies:

a = 1

p = 2

l = 1

e = 1