Assignment

Reg no:192519051

Shan roshan . I

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

IPO

* Input: A string
* Process: Count characters until null terminator
* Output: Length of the string

Program code:

#include <stdio.h>

Void main()

{

char str[100];

int i = 0;

printf("Enter a string: ");

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

{

i++;

}

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

}

Sample Output:

Enter a string: hello

Length of the string: 5

2) Write a program to copy one string to another.

IPO:

* Input: One string
* Process: Copy character by character
* Output: Copied string

Program code:

#include <stdio.h>

Void main()

{

char str1[100], str2[100];

int i;

printf("Enter a string to copy: ");

gets(str1);

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

{

str2[i] = str1[i];

}

str2[i] = '\0';

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

}

Sample output:

Enter a string to copy: world

Copied string: world

3) Write a program to concatenate two strings.

IPO:

* Input: Two strings
* Process: Append second string at end of first
* Output: Concatenated string

Program code:

#include <stdio.h>

Void 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];

i++; j++;

}

str1[i] = '\0';

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

}

Sample output:

Enter first string: Hello

Enter second string: World

Concatenated string: HelloWorld

4) Write a program to compare two strings.

IPO;

* Input: Two strings
* Process: Compare each character
* Output: Whether strings are equal or not

Program 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");

}

Sample output:

Enter first string: test

Enter second string: test

Strings are equal

5) Write a program to count vowels and consonants in a string.

IPO:

* Input: A string
* Process: Check each character
* Output: Count of vowels and consonants

Program code:

#include <stdio.h>

Void main()

{

char str[100];

int vowels = 0, consonants = 0, i;

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\nConsonants: %d\n", vowels, consonants);

}

Sample output:

Enter a string: Hello

Vowels: 2

Consonants: 3

6)Write a program to convert lowercase to uppercase and vice versa.

IPO:

* Input: A string
* Process: Use ASCII value to switch case
* Output: Case-converted string

Program 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] = str[i] - 32;

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

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

}

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

}

Sample output:

Enter a string: HelloWorld

Converted string: hELLOwORLD

7) Write a program to check if a string is palindrome.

IPO:

* Input: A string
* Process: Compare first and last characters moving inward
* Output: Whether string is palindrome

Program code:

#include <stdio.h>

Void main()

{

char str[100];

int i, len, flag = 0;

printf("Enter a string: ");

gets(str);

len = 0;

while (str[len] != '\0') len++;

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

if (str[i] != str[len - i - 1]) {

flag = 1;

break;

}

}

if (flag == 0)

printf("String is a palindrome\n");

else

printf("String is not a palindrome\n");

}

Sample output:

Enter a string: madam

String is a palindrome

8) Write a program to reverse a string

IPO:

* Input: A string
* Process: Swap characters from ends
* Output: Reversed string

Program code:

#include <stdio.h>

Void main()

{

char str[100], temp;

int i, len;

printf("Enter a string: ");

gets(str);

len = 0;

while (str[len] != '\0') len--;

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);

}

Sample output:

Enter a string: hello

Reversed string: olleh

9) Write a program to count words in a string.

IPO:

* Input: A sentence
* Process: Count spaces between words
* Output: Number of words

Program 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("Total words: %d\n", count);

}

Sample output:

Enter a string: This is a test

Total words: 4

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

IPO:

* Input: A string
* Process: Count occurrences of each character
* Output: Frequency of each character

Program 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]);

}

}

Sample output:

Enter a string: aabcc

Character frequencies:

a: 2

b: 1

c: 2