

1) strlen :-

strlen is a function in C that calculates the length of a null-terminated string.

Definition :- size_t strlen (const char *str.)

It returns the number of characters in the string, excluding the null character (\0) at the end.

Example :-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char str[] = "hello";
    printf("%d", strlen(str));
    return 0;
}
```

2) strcpy definition

strcpy is a function in C that copies a string from one location to another.

Definition :- char *strcpy(char *dest, const char *src)

It copies the string pointed to by src (including the null character \0) to the location pointed to by dest.

Example :-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char str[] = "hello";
    char dest[10];
    strcpy(dest, str);
    printf("%s", dest);
    return 0;
}
```

3) strcpy

specified

string +

char =

It cop

dest.

with

Exam

#in

int

char

dest

3) `strncpy` is a function in C that copies a specified number of characters from one string to another.

Definition: - `char *strncpy (char *dest, const char *src, size_t n)`

It copies at most n characters from `src` to `dest`. If `src` is shorter than n , it pads `dest` with null characters (10).

Example:-

```
#include <stdio.h>
#include <conio.h>
int main () {
    char src[] = "hello";
    char dest[10];
    strncpy(dest, src, 3);
    dest[3] = '\0'; // ensure null-terminal
    printf("copied string : %s\n", dest);
    return 0;
}
```

4) `strcat` is a function in C that concatenates two strings.

Definition: `char *strcat (char *dest, const char *src)`

It appends the string pointed to by `src` to the end of the string pointed to by `dest`, overwriting the null character (10) at the end of `dest`.

Example:

```
#include <stdio.h>
#include <conio.h>
int main () {
    char dest[20] = "hello";
    char src[] = "world";
    strcat(dest, src);
    printf("concatenated string : %s\n");
}
```

return 0;

1) 3

5) `strcat` is a function in C that concatenates a specified number of characters from one string to another.

Definition : `char *strcat (char *dest, const char *src, size_t n)`

It appends at most n characters from `src` to the end of `dest`, overwriting the null character at the end of `dest`, and adds a null character at the end of the concatenated string.

Example:-

```
#include <stdio.h>
#include <conio.h>
int main () {
    char dest [20] = "hello";
    char src [] = "world";
    strcat (dest, src, 3);
    printf ("concatenated string : %s\n", dest);
    return 0;
}
```

6) `strcmp` is a function in C that compares two strings.

Definition :- `int strcmp (const char *s1, const char *s2)`

It compares the strings `s1` and `s2` lexicographically (character by character).

Example:- `#include <stdio.h>`

```
#include <conio.h>
int main () {
    char s1 [] = "apple";
    char s2 [] = "banana";
    int result = strcmp (s1, s2);
    if (result < 0)
        printf ("%s is less than %s\n", s1, s2);
    else if (result > 0)
        printf ("%s is greater than %s\n", s1, s2);
    else
        printf ("%s is equal to %s\n", s1, s2);
}
```

```
printf ("%s\n");
else
    printf "%s\n";
3
```

7) `strcmp` is a function in C that compares two strings lexicographically.

It compares the strings `s1` and `s2` lexicographically.

Example:-

```
#include <stdio.h>
#include <conio.h>
int
```

```
    printf ("%os is equal to %s\n", s1, s2);  
else  
    printf ("%os is greater than %s\n", s1, s2);  
return 0;  
}
```

7) `strcmp` is a function in C that compares a specified number of characters from two strings.

Definition:- `int strcmp (const char *s1, const char *s2, size_t n)`

It compares at most `n` characters from `s1` and `s2` lexicographically (character by character).

Example:-

```
#include <stdio.h>  
#include <conio.h>  
int main () {  
    char s1[] = "apple";  
    char s2[] = "appoz";  
    int result = strcmp (s1, s2, 4);  
    if (result < 0)  
        printf ("%os is less than %os (up to 4 chars)  
                \n", s1, s2);  
    else if (result == 0)  
        printf ("%os is equal to %os (up to 4 chars)  
                \n", s1, s2);  
    else  
        printf ("%os is greater than %os (up to 4  
                chars)\n", s1, s2);  
    return 0;  
}
```