C String Functions <string.h>

Strings are sequences of characters stored in an array. The following functions are used to manipulate strings in C programming.

1. String Copy: strncpy()

This function copies a specified number of characters from one string to another.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char source[] = "Hello, World!";
   char destination[20];
   strncpy(destination, source, 5);
   destination[5] = '\0';  // Ensure null-termination
   printf("strncpy(): %s\n", destination);
   return 0;
}
```

Output:

```
strncpy(): Hello
```

2. String Concate: strncat()

This function concatenates a specified number of characters from the second string to the first.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char str1[20] = "Hello, ";
   char str2[] = "World!";
   strncat(str1, str2, 3);
   printf("strncat(): %s\n", str1);
   return 0;
}
```

Output:

```
strncat(): Hello, Wor
```

3. String Compare: strncmp()

This function compares a specified number of characters between two strings.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char str3[] = "apple";
   char str4[] = "appetizer";
   int result = strncmp(str3, str4, 4);
   printf("strncmp(): %d\n", result);
   return 0;

}
```

Output:

strncmp(): 7

4. First Character Occurrence: strchr()

This function locates the first occurrence of a character in a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char sentence[] = "This is a sample sentence.";
   char *ptr = strchr(sentence, 'a');
   printf("strchr(): %s\n", ptr);
   return 0;
}
```

Output:

strchr(): a sample sentence.

5. Last Character Occurrence: strrchr()

This function locates the last occurrence of a character in a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char sentence[] = "This is a sample sentence.";
   char *lastPtr = strrchr(sentence, 'a');
   printf("strrchr(): %s\n", lastPtr);
   return 0;
}
```

Output:

```
strrchr(): ample sentence.
```

6. String Search: strstr()

This function searches for the first occurrence of a substring within a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char text[] = "The quick brown fox jumps over the lazy dog.";
   char *substr = strstr(text, "fox");
   printf("strstr(): %s\n", substr);
   return 0;
}
```

Output:

```
strstr(): fox jumps over the lazy dog.
```

7. String Token Break: strtok()

This function breaks a string into a series of tokens based on a delimiter.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char sentence[] = "This is a sample sentence";
   char *token = strtok(sentence, " ");
   while (token != NULL) {
      printf("strtok(): %s\n", token);
      token = strtok(NULL, " ");
```

```
    }
    return 0;
}
```

Output:

strtok(): This

strtok(): is

strtok(): a

strtok(): sample

strtok(): sentence

8. Lowercase String:

By default there is no function in string.h that can convert a string to lower case. So for that we can create a user define function using ctype .h that can do the needful.

Example:

```
#include<stdio.h>
#include<ctype.h>

char lower(char *str){    //user_define function

    for(int i=0;str[i]!='\0';i++){
        str[i]=tolower(str[i]);
    }
}
int main()
{
    char str[ ] = "HELLO I AM";
    lower(str);
    printf("%s",str);

}
```

Output:

hello i am

9. Uppercase String:

By default there is no function in string.h that can convert a string to upper case. So for that we can create a user define function using ctype .h that can do the needful.

Example:

```
#include<stdio.h>
#include<ctype.h>

char upper(char *str){  //user_define function

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

int main()
{
   char str[ ] = "hello i am";
   upper(str);
   printf("%s",str);
}
```

Output:

```
HELLO I AM
```

10. Duplicate String: strdup()

This function duplicates a string.

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
   char original[] = "Original String";
   char *duplicate = strdup(original);
   printf("strdup(): %s\n", duplicate);
   return 0;
}
```

Output:

```
strdup(): Original String
```

11. Length of a String: strlen()

Example:

```
#include <stdio.h>
#include <string.h>

int main() {
  char alphabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  printf("%d", strlen(alphabet));
  return 0;
}
```

Output:

26

12. Concatenation of two Strings: strcat()

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
  char str1[20] = "Hello ";
  char str2[] = "World!";
  // Concatenate str2 to str1 (the result is stored in str1)
  strcat(str1, str2);
  // Print str1
  printf("%s", str1);
  return 0;
}
```

Output:

Hello World!

13. Copy Strings: strcpy()

Example:

```
#include <stdio.h>
#include <string.h>
int main() {
  char str1[20] = "Hello World!";
  char str2[20];
  // Copy str1 to str2
  strcpy(str2, str1);
```

```
  // Print str2
  printf("%s", str2);
  return 0;
}
```

14. Compare Strings: strcmp()

To compare two strings, you can use the strcmp() function.It returns 0 if the two strings are equal, otherwise a value that is not 0.

Example:

```
#include <stdio.h>
#include <string.h>

int main() {
  char str1[] = "Hello";
  char str2[] = "Hello";
  char str3[] = "Hi";

  // Compare str1 and str2, and print the result
  printf("%d\n", strcmp(str1, str2));

  // Compare str1 and str3, and print the result
  printf("%d\n", strcmp(str1, str3));

  return 0;
}
```

Output:

```
0
-4
```