In C, string functions are part of the standard library, primarily found in the <string.h> header file. Here's a list of commonly used string functions:

**1. String Manipulation Functions**

* **strcpy()**: Copies one string to another.
* char \*strcpy(char \*destination, const char \*source);
* **strncpy()**: Copies a specified number of characters from one string to another.
* char \*strncpy(char \*destination, const char \*source, size\_t num);
* **strcat()**: Appends one string to the end of another.
* char \*strcat(char \*destination, const char \*source);
* **strncat()**: Appends a specified number of characters from one string to another.
* char \*strncat(char \*destination, const char \*source, size\_t num);

**2. String Comparison Functions**

* **strcmp()**: Compares two strings.
* int strcmp(const char \*str1, const char \*str2);
* **strncmp()**: Compares up to a specified number of characters of two strings.
* int strncmp(const char \*str1, const char \*str2, size\_t num);

**3. String Length and Search Functions**

* **strlen()**: Returns the length of a string.
* size\_t strlen(const char \*str);
* **strchr()**: Finds the first occurrence of a character in a string.
* char \*strchr(const char \*str, int ch);
* **strrchr()**: Finds the last occurrence of a character in a string.
* char \*strrchr(const char \*str, int ch);
* **strstr()**: Finds the first occurrence of a substring in a string.
* char \*strstr(const char \*haystack, const char \*needle);

**4. String Tokenization**

* **strtok()**: Splits a string into tokens based on delimiters.
* char \*strtok(char \*str, const char \*delimiters);

**5. String Conversion Functions**

* **strtol()**: Converts a string to a long integer.
* long int strtol(const char \*str, char \*\*endptr, int base);
* **strtod()**: Converts a string to a double.
* double strtod(const char \*str, char \*\*endptr);
* **atoi()**: Converts a string to an integer (deprecated; use strtol() instead).
* int atoi(const char \*str);

**6. Memory Management Functions for Strings**

* **memcpy()**: Copies a block of memory.
* void \*memcpy(void \*destination, const void \*source, size\_t num);
* **memmove()**: Moves a block of memory, even if regions overlap.
* void \*memmove(void \*destination, const void \*source, size\_t num);
* **memset()**: Fills a block of memory with a specified value.
* void \*memset(void \*ptr, int value, size\_t num);

**Example Usage**

#include <stdio.h>

#include <string.h>

int main() {

char str1[20] = "Hello";

char str2[20] = "World";

// String concatenation

strcat(str1, str2);

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

// String comparison

int result = strcmp("abc", "abcd");

printf("Comparison Result: %d\n", result);

// String length

size\_t len = strlen(str1);

printf("Length of String: %zu\n", len);

return 0;

}

Let me know if you need detailed explanations or examples for any specific function!