

String Manipulation in C

String manipulation in C is handled using character arrays (char[]) or pointers to char since C does not have a built-in string type like some other languages (e.g., C++, Python). Strings in C are null-terminated, meaning the last character is always the null character (\0).

Techniques and Functions in String Manipulation

Here are some common string manipulation techniques and functions in C:

1. String Initialization

```
char str1[] = "Hello, World!";  
char str2[50] = "Hello, World!"; // Array of size 50, but string is 13 characters long.
```

2. Copying Strings

To copy a string from one array to another, you can use strcpy() from the string.h library:

```
#include <string.h>  
char str1[] = "Hello";  
char str2[20];  
strcpy(str2, str1); // Copies "Hello" to str2
```

3. Concatenating Strings

To concatenate two strings, use strcat():

```
char str1[50] = "Hello";  
char str2[] = ", World!";  
strcat(str1, str2); // str1 is now "Hello, World!"
```

4. String Length

To get the length of a string, use strlen():

```
int len = strlen(str1); // Returns length of the string "Hello, World!" (13)
```

5. Comparing Strings

To compare two strings, use strcmp():

```
int result = strcmp(str1, str2);  
if (result == 0) {  
    printf("Strings are equal.\n");  
} else {  
    printf("Strings are not equal.\n");  
}
```

6. Reversing a String

You can write a simple loop to reverse a string:

```
void reverse(char str[]) {  
    int len = strlen(str);  
    for (int i = 0; i < len / 2; i++) {
```

```

        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}

char str[] = "Hello, World!";
reverse(str); // Now str is "!dlroW ,olleH"

```

7. Splitting a String (Tokenizing)

To split a string by a delimiter, use strtok():

```

char str[] = "Hello,World,Split";
char *token = strtok(str, ",");
while (token != NULL) {
    printf("%s\n", token);
    token = strtok(NULL, ",");
}

```

8. Converting Case

You can convert the case of characters manually by iterating through the string:

```

#include <ctype.h>
for (int i = 0; str1[i] != '\0'; i++) {
    str1[i] = toupper(str1[i]); // Converts to uppercase
}

```

Example:

```

#include <stdio.h>
#include <string.h>
int main() {
    char str1[50] = "Hello";
    char str2[] = ", World!";
    // Concatenate strings
    strcat(str1, str2);
    printf("Concatenated: %s\n", str1);

    // Copy string
    char str3[50];
    strcpy(str3, str1);
    printf("Copied: %s\n", str3);

    // Get length
    int len = strlen(str1);
    printf("Length: %d\n", len);
}

```

```
// Compare strings
if (strcmp(str1, str3) == 0) {
    printf("Strings are equal.\n");
} else {
    printf("Strings are not equal.\n");
}

return 0;
}
```