

String Functions

You can create your functions to do many things with strings. For standard tasks, there are function in the `string` library.

1 Creating Your Own String Function

Almost all string functions have a main loop that iterates until the end of the string is detected. For example, here is code to compute the length of a string:

```
int strlen(char s[]) {
    int x = 0;
    while (s[x] != '\0')
        x=x+1;
    return x;
}
```

It returns 5 when called by

```
char test[] = "happy";
strlen(test);
```

Or suppose you wanted to convert a string to all capitals:

```
void toUpperCase(char s[]) {
    int x;
    for( x=0; s[x] != '\0'; x++ ) {
        if( s[x]>='a' && s[x]<='z')
            s[x] += 'A' - 'a';
    }
}
```

2 String Library

Further string functions are available in the `string` library. But beware! If the array you are copying to is not big enough, then crash: the null-character terminator gets lost. These functions include:

- `strcpy(dest,src)` copies one string into another
- `strcat(dest,src)` appends one string to another

- `strlen(s)` returns the string length
- `strcmp(first,second)` compares the strings alphabetically: it returns a negative value if the first comes before the second, 0 if the two strings are the same, and a positive value if the first comes after the second.

The only one you have to know is `strlen`.

3 Example Program: `stringLibrary.c`

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

int main(void) {

    char alpha[] = "alpha";
    char beta[] = "beta";
    char final[100];

    strcpy(final, alpha);
    if( strcmp(final,alpha)==0 ) // true
        printf("%s and %s are the same\n", final, alpha );

    strcat(final,beta); // final is now "alphabetabeta"
    if( strcmp(final,alpha)>0 ) // true
        printf("%s comes after %s\n", final, alpha );

    return 0;
}
```