

C Function Definitions

```
size_t strlen(const char *s);
```

The `strlen()` function calculates the length of the string pointed to by `s`, excluding the terminating null byte (`'\0'`).

```
int strncmp(const char *s1, const char *s2, size_t n);
```

The `strncmp()` function compares the first (at most) `n` bytes of two strings `s1` and `s2`. It returns an integer less than, equal to, or greater than zero if `s1` is found, respectively, to be less than, to match, or be greater than `s2`.

```
char *strncpy(char *dest, const char *src, size_t n);
```

The `strncpy()` function copies the string pointed to by `src`, including the terminating null byte (`'\0'`), to the buffer pointed to by `dest`. The strings may not overlap, and at most `n` bytes of `s` are copied. Warning: If there is no null byte among the first `n` bytes of `src`, the string placed in `dest` will not be null-terminated.

If the length of `src` is less than `n`, `strncpy()` writes additional null bytes to `dest` to ensure that a total of `n` bytes are written.

```
char *gets(char *s);
```

`gets()` reads a line from `stdin` into the buffer pointed to by `s` until either a terminating newline or EOF, which it replaces with a null byte (`'\0'`).

```
char *fgets(char *s, int size, FILE *stream);
```

`fgets()` reads in at most one less than `size` characters from `stream` and stores them into the buffer pointed to by `s`. Reading stops after an EOF or a newline. If a newline is read, it is stored into the buffer. A terminating null byte (`'\0'`) is stored after the last character in the buffer.