

## Mahdollisesti hyödyllisiä funktioita

### Merkkijonojen käsittelyyn (määritelty string.h - otsakkeessa):

- `size_t strlen(const char *s)`; palauttaa annetun merkkijonon **s** pituuden.
- `char *strcpy(char *dest, const char *src)`; kopioi merkkijonon **src** paikkaan **dest**
- `char *strncpy(char *dest, const char *src, size_t n)`; kopioi enintään **n** merkkiä merkkijonosta **src** merkkijonoon **dest**. Jos merkkijono on lyhyempi kuin **n**, loput tavut täytetään `'\0'` - merkillä.
- `char *strcat(char *dest, const char *src)`; liittää merkkijonon **src** merkkijonon **dest** perään
- `char *strncat(char *dest, const char *src, size_t n)`; liittää enintään **n** merkkiä merkkijonosta **src** merkkijonoon **dest**
- `int strcmp(const char *s1, const char *s2)`; palauttaa 0 jos annetut merkkijonot ovat samat, erisuuri kuin 0 jos merkkijonot eroavat

### Muistinhallinta (määritelty stdlib.h - otsakkeessa, memset string.h:ssa):

- `void *malloc(size_t size)`; Varaa **size** tavua muistia, palauttaa osoitteen varattuun muistialueeseen
- `void *calloc(size_t nmemb, size_t size)`; Varaa **nmemb** kertaa **size** tavua muistia, nolaa varatun muistialueen
- `void *realloc(void *ptr, size_t size)`; Muuttaa muistialueen **ptr** koon **size**:ksi, palauttaa uuden osoittimen muistialueeseen
- `void free(void *ptr)`; vapauttaa annetun muistialueen
- `void *memset(void *s, int c, size_t n)`; asettaa muistialueen **s**, jonka koko on **n**, kaikki tavut **c**:ksi

### Merkkien käsittelyyn (määritelty ctype.h - otsakkeessa):

- `int toupper(int c)`; muuta merkki isoksi kirjaimeksi
- `int tolower(int c)`; muuta merkki pieneksi kirjaimeksi
- `int isalnum(int c)`; onko merkki kirjain tai numero?
- `int isalpha(int c)`; onko merkki kirjain?
- `int isspace(int c)`; onko merkki tyhjä väli?
- `int islower(int c)`; onko merkki pieni kirjain?
- `int isupper(int c)`; onko merkki iso kirjain?

### Muotoiltu I/O (määritelty stdio.h - otsakkeessa):

- `int printf(const char *format, ...)`; tulostaa muotoiltua tulostetta annetusta merkkijonosta ja parametreista
- `int scanf(const char *format, ...)`; lukee muotoiltua syötettä annettuihin osoitteisiin. Parametrit ovat siis muistiosoitteita

### Muotoilumääreitä printf- ja scanf-funktioihin:

- `%d`: kokonaisluku
- `%f`: liukuluku
- `%u`: etumerkitön kokonaisluku
- `%x`: heksadesimaaliluku
- `%c`: merkki
- `%s`: merkkijono

### Lukuja binäärimuodossa

- |   |                         |                         |
|---|-------------------------|-------------------------|
| • | <code>0x0</code> : 0000 | <code>0x8</code> : 1000 |
| • | <code>0x1</code> : 0001 | <code>0x9</code> : 1001 |
| • | <code>0x2</code> : 0010 | <code>0xA</code> : 1010 |
| • | <code>0x3</code> : 0011 | <code>0xB</code> : 1011 |
| • | <code>0x4</code> : 0100 | <code>0xC</code> : 1100 |
| • | <code>0x5</code> : 0101 | <code>0xD</code> : 1101 |
| • | <code>0x6</code> : 0110 | <code>0xE</code> : 1110 |
| • | <code>0x7</code> : 0111 | <code>0xF</code> : 1111 |