

nützliche Funktionen

Im Folgenden sind einige c-Funktionen aufgelistet, die bei der Bearbeitung der Hausaufgaben nützlich sein können. Details zur Nutzung und weitere library-Funktionen findet ihr im Internet.

Funktion	kurze Beschreibung
<code>char *strtok(char *str, const char *delim)</code>	unterteilt die Zeichenfolge str mithilfe des Trennzeichens delim in mehrere Token
<code>int strcmp(const char* str1, const char* str2)</code>	gibt 0 zurück, wenn zwei Strings gleich sind
<code>char *strchr (char *str, int character)</code>	gibt einen Zeiger auf das letzte Vorkommen des Zeichens im String str zurück (Es wird typischerweise ein char übergeben, auch wenn int character als Parameter steht.)
<code>char *strdup(const char *str1)</code>	gibt einen Zeiger auf einen nullterminierten String zurück, der ein Duplikat der Zeichenfolge ist, auf die str1 zeigt
<code>char *strncpy (char *destination, const char *source, size_t num)</code>	kopiert die ersten num zeichen von source in das Ziel destination
<code>char *strcat(char *dest, const char *src)</code>	hängt den String src am Ende von dest an
<code>void* memcpy(void* dest, const void* src, std::size_t count)</code>	kopiert count Bytes von dem src-Objekt in das dest-Objekt
<code>void * memset (void *ptr, int value, size_t num)</code>	setzt die ersten num Bytes des Speicherblocks auf den ptr zeigt auf value
<code>void *realloc(void *ptr, size_t size)</code>	versucht, die Größe des Speicherblocks zu ändern, auf den ptr zeigt und der zuvor mit einem Aufruf von malloc oder calloc zugewiesen wurde
<code>FILE *fopen(const char *filename, const char *mode)</code>	öffnet eine Datei und gibt einen Pointer auf diese zurück. Mit mode wird die Zugriffsart (lesen, schreiben, ...) angegeben.
<code>size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream)</code>	liest Daten aus dem angegebenen Stream in das Array auf das ptr zeigt
<code>size_t fwrite (const void *ptr, size_t size, size_t count, FILE *stream)</code>	schreibt ein Array mit count Elementen der Größe size aus dem ptr-Speicherblock an die aktuelle Position in stream
<code>int fclose(FILE *stream)</code>	schließt den Stream