

文字列 3

文字列処理

文字列処理のためのライブラリ関数がC言語には用意されている。以下の表に示す。

表 2: 文字列処理関数。#include <string.h>が必要。変数は、char s1[256], s2[256];のように文字型の配列である。配列のサイズは処理に必要なサイズよりも大きいこと(256 とは限らない)。c は文字型の変数 char c; である。

関数名	動作
strlen(s1)	文字列 s1 の長さ、すなわち文字数を整数値として返す。
strcpy(s1, s2)	s1 に、文字列 s2 をコピーする。
strcat(s1, s2)	文字列 s1 の後に、文字列 s2 をコピーする。
strcmp(s1, s2)	文字列 s1 と s2 を比較する。 s1 > s2 の場合、戻り値は正 s1 == s2 の場合、戻り値は 0 s1 < s2 の場合、戻り値は負
strncpy(s1, s2, n)	s1 に文字列 s2 の先頭から n 文字をコピーする。
strncat(s1, s2, n)	文字列 s1 の後に文字列 s2 の先頭から n 文字を連結する。
strncmp(s1, s2, n)	文字列 s1 と文字列 s2 の先頭から n 文字を比較する。比較の結果は、strcmp と同じ。
strchr(s1, c)	文字列 s1 の中の文字 c の位置を整数で返す。文字がないときは、NULL を返す。
strstr(s1, s2)	文字列 s1 中にある文字列 s2 の位置を整数で返す。もし、文字列がない場合、NULL を返す。

課題 2 の終了者は上記の関数と同じ仕様の関数を作成する。例えば、関数 strlen ならば str_length という関数名で作成する。教科書 p.256 以降には str_copy、str_length、str_cat、str_comp の例があるので、どのように作成しているかを確認し、それ以外を選択すること。