

חומר עזר

פונקציות לטיפול תווים

Prototype	Description
<code>int isdigit(int c);</code>	Returns true if c is a digit and false otherwise.
<code>int isalpha(int c);</code>	Returns true if c is a letter and false otherwise.
<code>int isalnum(int c);</code>	Returns true if c is a digit or a letter and false otherwise.
<code>int isxdigit(int c);</code>	Returns true if c is a hexadecimal digit character and false otherwise.
<code>int islower(int c);</code>	Returns true if c is a lowercase letter and false otherwise.
<code>int isupper(int c);</code>	Returns true if c is an uppercase letter; false otherwise.
<code>int tolower(int c);</code>	If c is an uppercase letter, tolower returns c as a lowercase letter. Otherwise, tolower returns the argument unchanged.
<code>int toupper(int c);</code>	If c is a lowercase letter, toupper returns c as an uppercase letter. Otherwise, toupper returns the argument unchanged.
<code>int isspace(int c);</code>	Returns true if c is a white-space character—newline ('\n'), space (' '), form feed ('\f'), carriage return ('\r'), horizontal tab ('\t'), or vertical tab ('\v')—and false otherwise
<code>int iscntrl(int c);</code>	Returns true if c is a control character and false otherwise.
<code>int ispunct(int c);</code>	Returns true if c is a printing character other than a space, a digit, or a letter and false otherwise.
<code>int isprint(int c);</code>	Returns true value if c is a printing character including space (' ') and false otherwise.
<code>int isgraph(int c);</code>	Returns true if c is a printing character other than space (' ') and false otherwise.

כמה פונקציות על מחרוזות

Function prototype	Function description
<code>char *strncpy(char *s1, const char *s2, size_t n)</code>	Copies at most n characters of string s2 into array s1. Does not copy '\0'. The value of s1 is returned.
<code>char *strcat(char *s1, const char *s2)</code>	Appends string s2 to array s1. The first character of s2 overwrites the terminating null character of s1. The value of s1 is returned.
<code>char *strncat(char *s1, const char *s2, size_t n)</code>	Appends at most n characters of string s2 to array s1. The first character of s2 overwrites the terminating null character of s1. Places '\0' after last character. The value of s1 is returned.

Function prototype	Function description
<code>char *strchr(const char *s, int c);</code>	Locates the first occurrence of character c in string s. If c is found, a pointer to c in s is returned. Otherwise, a NULL pointer is returned.
<code>size_t strcspn(const char *s1, const char *s2);</code>	Determines and returns the length of the initial segment of string s1 consisting of characters not contained in string s2.
<code>size_t strspn(const char *s1, const char *s2);</code>	Determines and returns the length of the initial segment of string s1 consisting only of characters contained in string s2.
<code>char *strpbrk(const char *s1, const char *s2);</code>	Locates the first occurrence in string s1 of any character in string s2. If a character from string s2 is found, a pointer to the character in string s1 is returned. Otherwise, a NULL pointer is returned.
<code>char *strrchr(const char *s, int c);</code>	Locates the last occurrence of c in string s. If c is found, a pointer to c in string s is returned. Otherwise, a NULL pointer is returned.
<code>char *strstr(const char *s1, const char *s2);</code>	Locates the first occurrence in string s1 of string s2. If the string is found, a pointer to the string in s1 is returned. Otherwise, a NULL pointer is returned.

פונקציות לשינוי ומציאת המיקום בקובץ

rewind(מצביע לקובץ)

מזיזה את המיקום הנוכחי בקובץ (file position pointer) לתחילתו.

ftell(מצביע לקובץ)

מחזירה את המיקום הנוכחי בקובץ.

fseek(מצביע לקובץ, offset, origin)

מזיזה את המיקום הנוכחי בקובץ:

- offset הוא מספר הבתים שיש לזוז (יכול להיות גם שלילי).
- origin הוא מוצא התזוזה (אחד משלושה):
 1. SEEK_CUR - המיקום הנוכחי
 2. SEEK_SET - תחילת הקובץ
 3. SEEK_END - סוף הקובץ

פונקציות לקלט ופלט לקבצים בינאריים

`size_t fread(void *buffer, size_t size, size_t count, FILE *stream)`

- קוראת נתונים מהקובץ
- `buffer` – מצביע לבלוק הזיכרון
- `size` – גודל בבתים של כל איבר לקריאה
- `count` – מספר האיברים לקריאה, כל אחד בגודל `size`
- `stream` – מצביע לקובץ
- מחזירה את מספר האיברים שנקראו בהצלחה

`size_t fwrite(const void *buffer, size_t size, size_t count, FILE *stream)`

- כותבת נתונים לקובץ
- `buffer` – מצביע למערך הנתונים להיכתב
- `size` – גודל בבתים של כל איבר להיכתב
- `count` – מספר האיברים להיכתב, כל אחד בגודל `size`
- `stream` – מצביע לקובץ
- מחזירה את מספר האיברים שנכתבו בהצלחה