

Biblioteca de Manipulación de Caracteres

Funciones de manipulación de caracteres

Prototipo	Descripción de la función
<code>int isdigit(int c);</code>	Devuelve un valor verdadero si c es un dígito de lo contrario devuelve 0 (falso).
<code>int isalpha(int c);</code>	Devuelve un valor verdadero si c es una letra de lo contrario devuelve 0 (falso).
<code>int isalnum(int c);</code>	Devuelve un valor verdadero si c es un dígito o una letra de lo contrario devuelve 0 (falso).
<code>int isxdigit(int c);</code>	Devuelve un valor verdadero si c es un dígito hexadecimal de lo contrario devuelve 0 (falso). (Revise el apéndice E, Sistemas de numeración, para una explicación detallada acerca de los números binarios, números octales, números decimales y números hexadecimales.)
<code>int islower(int c);</code>	Devuelve un valor verdadero si c es una letra minúscula de lo contrario devuelve 0 (falso).
<code>int isupper(int c);</code>	Devuelve un valor verdadero si c es una letra mayúscula de lo contrario devuelve 0 (falso).
<code>int tolower(int c);</code>	Si c es una letra mayúscula, tolower devuelve c como una letra minúscula. De lo contrario, tolower devuelve el argumento sin modificación.
<code>int toupper(int c);</code>	Si c es una letra minúscula, toupper devuelve c como una letra mayúscula. De lo contrario, toupper devuelve el argumento sin modificación.
<code>int isspace(int c);</code>	Devuelve un valor verdadero si c es un carácter de espacio en blanco (nueva línea ('\n'), espacio (' '), avance de página ('\f'), retorno de carro ('\r'), tabulador horizontal ('\t') o tabulador vertical ('\v') de lo contrario devuelve 0.
<code>int iscntrl(int c);</code>	Devuelve un valor verdadero si c es un carácter de control de lo contrario devuelve 0 (falso).
<code>int ispunct(int c);</code>	Devuelve un valor verdadero si c es un carácter de impresión diferente de un espacio, un dígito o una letra de lo contrario devuelve 0.
<code>int isprint(int c);</code>	Devuelve un valor verdadero si c es un carácter de impresión, incluso el espacio (' ') de lo contrario devuelve 0.
<code>int isgraph(int c);</code>	Devuelve un valor verdadero si c es un carácter de impresión diferente del espacio (' ') de lo contrario devuelve 0.

Funciones de manipulación de cadenas

Prototipo de la función	Descripción de la función
<code>char *strcpy(char *s1, const char *s2)</code>	Copia la cadena s2 dentro del arreglo s1. Devuelve el valor de s1.
<code>char *strncpy(char *s1, const char *s2, size_t n)</code>	Copia al menos n caracteres de la cadena s2 dentro del arreglo s1. Devuelve el valor de s1.
<code>char *strcat(char *s1, const char *s2)</code>	Agrega la cadena s2 al arreglo s1. El primer carácter de s2 sobrescribe al carácter de terminación nulo de s1. Devuelve el valor de s1.
<code>char *strncat(char *s1, const char *s2, size_t n)</code>	Agrega al menos n caracteres de la cadena s2 al arreglo s1. El primer carácter de s2 sobrescribe al carácter de terminación nulo de s1. Devuelve el valor de s1.

Funciones de comparación de cadenas

Prototipo de función	Descripción de la función
<code>int strcmp(const char *s1, const char *s2);</code>	Compara la cadena s1 con la cadena s2. La función devuelve 0, menor que 0, o mayor que 0, si s1 es igual, menor, o mayor que s2, respectivamente.
<code>int strncmp(const char *s1, const char *s2, size_t n);</code>	Compara hasta n caracteres de la cadena s1 con la cadena s2. La función devuelve 0, menor que 0, o mayor que 0, si s1 es igual, menor, o mayor que s2, respectivamente.

Funciones de conversión de cadenas

Prototipo de la función	Descripción de la función
<code>double atof(const char *ptrN);</code>	Convierte la cadena ptrN a double.
<code>int atoi(const char *ptrN);</code>	Convierte la cadena ptrN a int.
<code>long atol(const char *ptrN);</code>	Convierte la cadena ptrN a long int.
<code>double strtod(const char *ptrN, char **ptrFinal);</code>	Convierte la cadena ptrN a double.
<code>long strtol(const char *ptrN, char **ptrFinal, int base);</code>	Convierte la cadena ptrN a long.
<code>unsigned long strtoul(const char *ptrN, char **ptrFinal, int base);</code>	Convierte la cadena ptrN a unsigned long.

Funciones de entrada/salida de caracteres y cadenas

Prototipo de la función	Descripción de la función
<code>int getchar(void);</code>	Lee el siguiente carácter de la entrada estándar y lo devuelve como un entero.
<code>char *gets(char *s);</code>	Lee el siguiente carácter de la entrada estándar y lo coloca en el arreglo s hasta que encuentra un carácter de nueva línea o de fin de archivo. Agrega un carácter de terminación nulo al arreglo.
<code>int putchar(int c);</code>	Imprime el carácter almacenado en c.
<code>int puts(const char *s);</code>	Imprime la cadena s seguida por el carácter de nueva línea.
<code>int sprintf(char *s, const char *formato, ...);</code>	Equivalente a printf, excepto que la salida se almacena en el arreglo s, en lugar de imprimirse en la pantalla.
<code>int sscanf(char *s, const char *formato, ...);</code>	Equivalente a scanf, excepto que la entrada se lee desde el arreglo s, en lugar de leerlo desde el teclado.

Funciones de búsqueda

Prototipo de función	Descripción de la función
<code>char *strchr(const char *s, int c);</code>	Localiza la primera ocurrencia del carácter c en la cadena s. Si se localiza a c, se devuelve un apuntador a c en s. De lo contrario, se devuelve un apuntador NULL.
<code>size_t strcspn(const char *s1, const char *s2);</code>	Determina y devuelve la longitud del segmento inicial de la cadena s1, que consiste en los caracteres no contenidos en la cadena s2.
<code>size_t strspn(const char *s1, const char *s2);</code>	Determina y devuelve la longitud del segmento inicial de la cadena s1, que consiste sólo en los caracteres contenidos en la cadena s2.
<code>char *strpbrk(const char *s1, const char *s2);</code>	Localiza la primera ocurrencia en la cadena s1 de cualquier carácter de la cadena s2. Si se localiza un carácter de la cadena s2, se devuelve un apuntador al carácter de la cadena s1. De lo contrario, se devuelve un apuntador NULL.
<code>char *strrchr(const char *s, int c);</code>	Localiza la última ocurrencia de c en la cadena s. Si se localiza a c, se devuelve un apuntador a c en la cadena s. De lo contrario, se devuelve un apuntador NULL.
<code>char *strstr(const char *s1, const char *s2);</code>	Localiza la primera ocurrencia en la cadena s1 de la cadena s2. Si se localiza la cadena, se devuelve un apuntador a la cadena en s1. De lo contrario, se devuelve un apuntador NULL.
<code>char *strtok(char *s1, const char *s2);</code>	Una secuencia de llamadas a strtok separa la cadena s1 en "tokens" (piezas lógicas como palabras de una línea de texto) separados por caracteres contenidos en la cadena s2. La primera llamada contiene s1 como el primer argumento, y las llamadas subsiguientes contienen a NULL como el primer argumento, para continuar separando la misma cadena. Un apuntador al token actual es devuelto por cada llamada. Si no hay más tokens cuando se llama a la función, se devuelve NULL.

Referencias

- DEITEL H, DEITEL P "Cómo programar en C/C++ y Java" 4a Edición, Pearson Educación, México, 2004