



Referencia Rápida - Librería Estándar C (1)

Memoria - stdlib.h

void* malloc(size_t num_bytes)

Asigna una zona de memoria. Si no es posible, devuelve NULL

void* realloc(void* ptr, size_t num_bytes)

Modifica el tamaño de la zona de memoria a la que apunta ptr, haciendo que sea num_bytes. Devuelve un puntero a la nueva zona de memoria, o NULL si no se pudo redimensionar.

void free(void* ptr)

Libera una zona de memoria previamente asignada por calloc o malloc

void* calloc(size_t num, size_t tamaño)

Asigna una zona de memoria apropiada para num elementos, de tamaño bytes cada uno. Devuelve un puntero a la zona, o NULL si no hay memoria.

Cadenas - string.h

char* strcat(char* s1, char* s2)

char* strncat(char* s1, char *s2, size_t max)

Concatena s2 a s1. Devuelve un puntero a s1. No se permiten solapamientos

char* strcpy(char* s1, char* s2)

char* strncpy(char* s1, char *s2, size_t max)

Copia s2 sobre s1. Devuelve un puntero a s1. No se permiten solapamientos

int strcmp(char* s1, char* s2)

int strncmp(char* s1, char *s2, size_t max)

Devuelve >0 si s1 > s2, 0 si son iguales, <0 si s1 < s2.

int strlen(char* str)

Devuelve la longitud de una cadena. (En la longitud no se incluye el \0 final)

char* strchr(char* s, int c)

Devuelve un puntero a la primera posición de c en s, o NULL si no existe

char* strrchr(char* s, int c)

Devuelve un puntero a la última posición de c en s, o NULL si no existe

char* strstr(char* s1, char* s2)

Devuelve un puntero a la primera posición de s2 en s1, o NULL si no existe

Varias - stdlib.h

int rand()

Devuelve un valor aleatorio entre 0 y RAND_MAX

void srand(unsigned int semilla)

Inicializa el generador de números aleatorios con la semilla especificada

void qsort(void* datos, size_t numElem, size_t longElem, int (*comp)(void*,void*))

Ordenación QuickSort. El puntero datos apunta al inicio de los mismos, numElem es el número de datos, longElem el tamaño en bytes de cada uno y comp - un puntero a una función de comparación

void* bsearch(void* k, void* datos, size_t numElem, size_t longElem, int (*comp)(void*,void*))

Realiza una búsqueda binaria de k dentro de una matriz de datos que comienza en datos. La matriz debe estar ordenada de forma ascendente. Devuelve un puntero al dato encontrado o NULL si no se encuentra. En caso de duplicados cuál de ellos se devuelve es indeterminado.

char* getenv(char* nombre)

Devuelve el valor de una variable de entorno, o NULL si no existe.

int system(char* comando)

Ejecuta el comando especificado y devuelve su código de salida

Conversiones - stdlib.h

double atof(char* ptr)

int atoi(char* ptr)

long atol(char *ptr)

Convierten la cadena a la que apunta ptr a un número del tipo apropiado. La conversión se detiene en el primer carácter no válido. En caso de fallo, se devuelve 0.

double strtod(char* str, char fin)**

long strtol(char* str, char fin, int base)**

unsigned long strtoul(char* str, char* fin, int base)

Convierten la cadena a la que apunta ptr a un número del tipo apropiado. La conversión se detiene en el primer carácter no válido, y la posición de este carácter queda almacenada en el puntero fin. El parámetro base indica la base numérica (10, 16, 8, etc.) a utilizar para la conversión

Zonas de memoria - string.h

void* memchr(void* ptr, int c, size_t max)

Devuelve un puntero a la primera posición de c en ptr, o NULL si no existe

void* memcpy(void*s1,void* s2, size_t max)

Copia s2 encima de s1. Devuelve s1. No se permiten solapamientos

void* memmove(void*s1,void* s2, size_t max)

Copia s2 encima de s1. Devuelve s1. Se permiten solapamientos

void* memset(void*s1,int c, size_t max)

Rellena una zona de memoria con el byte c

Caracteres - ctype.h

la función...

...devuelve un valor != 0 (verdadero) para:

int isalnum(int c)

alfanuméricos (A..Z,a..z,0..9)

int isalpha(int c)

letras (A..Z,a..z)

int iscntrl(int c)

caracteres de control (00 a 31 y 126)

int isdigit(int c)

dígitos (0..9)

int isgraph(int c)

car. visibles (todos menos 0 a 31 y espacio)

int islower(int c)

minúsculas (a..z)

int isprint(int c)

car. imprimibles (todos menos 0 a 31)

int isspace(int c)

blancos (espacio, tabulador, etc.)

int isupper(int c)

mayúsculas (A..Z)

int isxdigit(int c)

dígitos en hexadecimal (0..9 y A..F o a..f)

int toupper(int c)

Convierte c a mayúsculas

int tolower(int c)

Convierte c a minúsculas

Funciones Matemáticas - math.h

Salvo indicación contraria, todos los parámetros son double

double exp(x)

e^x

double log(x)

$\ln(x)$

double pow(x,y)

x^y

double sqrt(x)

\sqrt{x}

double ceil(x)

redondeo dcha.

double floor(x)

redondeo izda.

sin(x)

cos(x)

Funciones trigonométricas

asin(x)

acos(x)

Funciones trigonométricas inversas

sinh(x)

cosh(x)

Funciones hiperbólicas

