

C Standard Library

Esta librería se encarga de la funcionalidad básica del lenguaje, como el manejo de memoria y los tipos de variables.

Para usarla debes escribir `#include <stdlib.h>` al principio de tu programa.

Tipos

`size_t`

Un entero positivo con la capacidad suficiente como para guardar la dirección de memoria más alta del computador.

Constantes

`NULL`

Representa una dirección de memoria inexistente. Esta es lógicamente equivalente a 0, y provocará que tu programa se caiga si tratas de leer a donde apunta.

Funciones

`size_t sizeof(type)`

Entrega el tamaño en bytes que necesita un determinado tipo para ser guardado en memoria.

```
sizeof(int) // Entrega el tamaño en bytes de un int

sizeof(struct lista) // Entrega el tamaño en bytes de un struct lista
```

`void* malloc(size_t size) // Memory Allocate`

Solicita `size` bytes de memoria contigua al sistema operativo, y entrega la dirección de memoria donde esta comienza. En caso de fallar, retorna `NULL`. La memoria viene tal cual como la haya dejado otro proceso.

```
Persona* p = malloc(sizeof(Persona)); // Ahora en p se puede guardar una persona

Persona** p = malloc(10 * sizeof(Persona*)); // Ahora en p se pueden guardar 10 personas
```

`void* calloc(int length, size_t size) // Clear Allocate`

Solicita (`size * length`) bytes de memoria contigua al sistema operativo, y entrega la dirección de memoria donde esta comienza. En caso de fallar, retorna `NULL`. La memoria viene con todos sus bytes en 0.

```
Persona* p = calloc(1, sizeof(Persona)); // Ahora en p se puede guardar una persona

Persona** p = calloc(10, sizeof(Persona*)); // Ahora en p se pueden guardar 10 personas
```

`void free(void* pointer)`

Indica al sistema operativo que la memoria solicitada ya no se usará más, para que pueda recuperarla.

```
Persona* p = malloc(sizeof(Persona));
/* ... */
free(p);
```

C Standard Input/Output

Esta librería se encarga del manejo de **input** y **output**, ya sea con archivos o a través de la consola.

Para usarla debes escribir `#include <stdio.h>` al principio de tu programa.

Tipos

FILE

Representa un *stream* sobre el cual se puede leer o escribir, ya sea la consola o un archivo en disco.

Constantes

FILE* stdout // *Standard Output*

Canal de salida de tu programa hacia la consola

FILE* stdin // *Standard Input*

Canal de entrada de la consola hacia tu programa

FILE* stderr // *Standard Error*

Canal secundario de salida de tu programa hacia la consola, usado para comunicar errores

Funciones

FILE* fopen(char* filename, char* mode) // *File open*

Abre el *stream* de el archivo, en el modo especificado.

```
FILE* fr = fopen("archivo.txt", "r") // Abre el archivo en modo de lectura
```

```
FILE* fw = fopen("archivo.txt", "w") // Abre el archivo en modo de escritura
```

fprintf(**FILE*** stream, char* format, ...) // *File print function*

Imprime el texto al *stream* especificado. Esta función puede recibir una múltiple cantidad de parámetros, especificada por el patrón en format

```
fprintf(fw, "La respuesta es %i", 42)
```

```
fprintf(fw, "Bienvenido, %s %s", nombre, apellido)
```

fscanf(**FILE*** stream, char* format, ...) // *File scan function*

Lee del *stream* a las variables, siguiendo el formato especificado. Requiere un puntero a las variables para poder escribir sobre ellas.

```
fscanf(fr, "La matriz era de %ix%i", &height, &width)
```

```
printf(char* format, ...)  $\cong$  fprintf(stdout, char* format, ...)
```

```
scanf(char* format, ...)  $\cong$  fscanf(stdin, char* format, ...)
```

fclose(**FILE*** stream) // *File close*

Indica al sistema operativo que el archivo ya no está en uso, y libera su memoria.