**STATIC allocation:**  is an allocation procedure used for the allocation of data object at compile time. The compiler decides the extent of storage which cannot be changed during runtime so it is easy for the processor to know the addresses of these data objects at a later stage.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

* Static and global variable are store at compile time and exists for the entire duration of the program
* Local variables are pushed/popped to the stack. Memory is automatically deallocated.
* Problem is that STACK can grow in size uncontrollably (see recursive functions)

jImmagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteImmagine che contiene testo, calligrafia, linea, Carattere

Descrizione generata automaticamente

**HEAP allocation:** Heap allocation is an allocation procedure in which the heap is used to manage the allocation of memory. Heap allocation overcomes the limitation of stack allocation. It is possible to retain the value of variables even after the activation record in heap allocation strategy which is not possible in stack allocation. It maintains a linked list for the free blocks and reuses the deallocated space using the best fit.

**STACK MEMORY**

**Purpose:** The stack is used for static memory allocation, which includes function calls, local variables, and control flow (like function return addresses)

**Characteristics:**

* Limited and fixed size
* Very fast (LIFO or FIFO)
* Automatically allocated/deallocated
* Variables stored on the stack are accessible only within the scope in which they are defined.
* Used for storing function calls, local variables, and control flow data.

**HEAP ALLOCATION**

**Purpose:** The heap is used for dynamic memory allocation, which is memory that is allocated during the runtime of the program and is not automatically managed.

**Characteristics:**

* Larger and more flexible than stack
* Lower speed due to more complex memory management
* Manual allocation/deallocation
* Variables or objects stored in heap memory can be accessed globally or passed around different parts of the program.
* Used for objects, data structures, and arrays whose size or lifetime is not known at compile time.