

Dynamic Memory Allocation in C

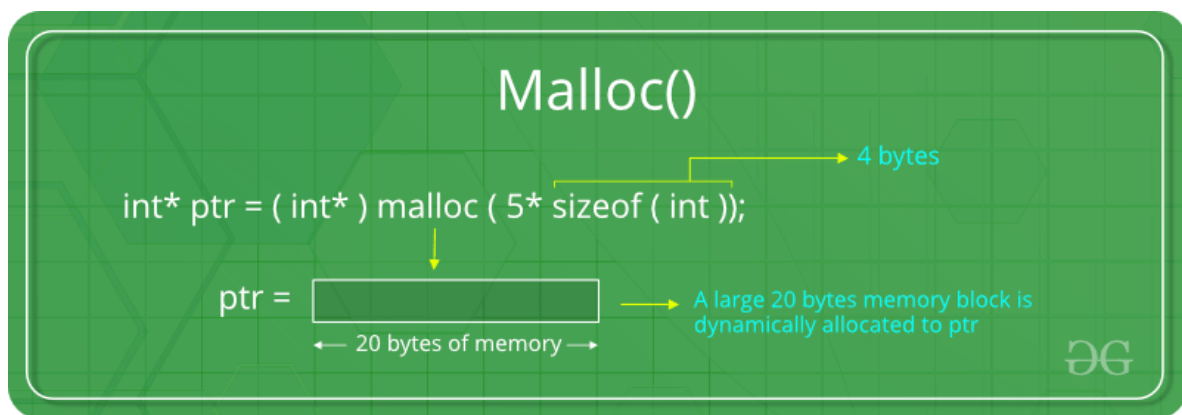
Braulio Millan Chin – Rob2A – July 5, 2021

Dynamic Memory Allocation is a procedure in which the size of a data structure (e.g. an Array) is changed during the program.

C provides some functions to achieve this. There are 4 functions provided by the <stdlib.h> library.

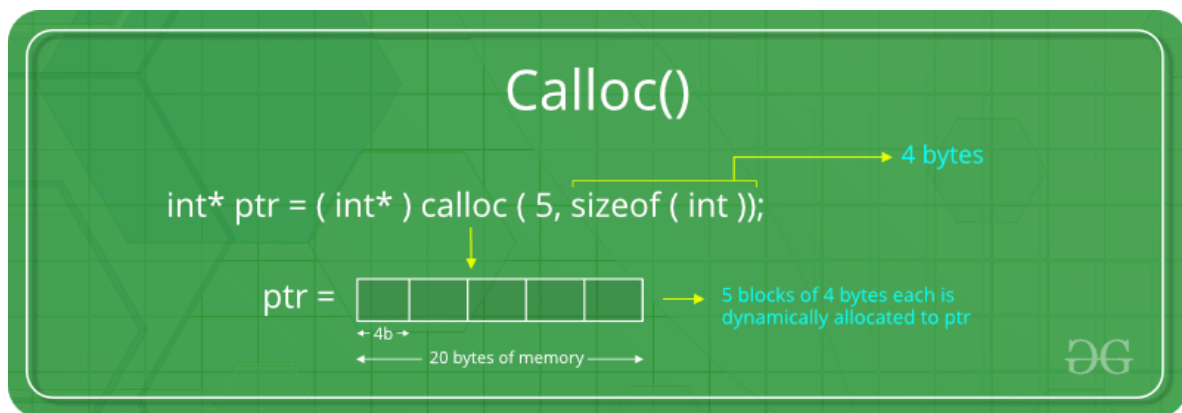
Malloc()

“Memory allocation”. It’s used to allocate a single large block of memory with the specified size. If space is insufficient, allocation fails and returns a NULL pointer.



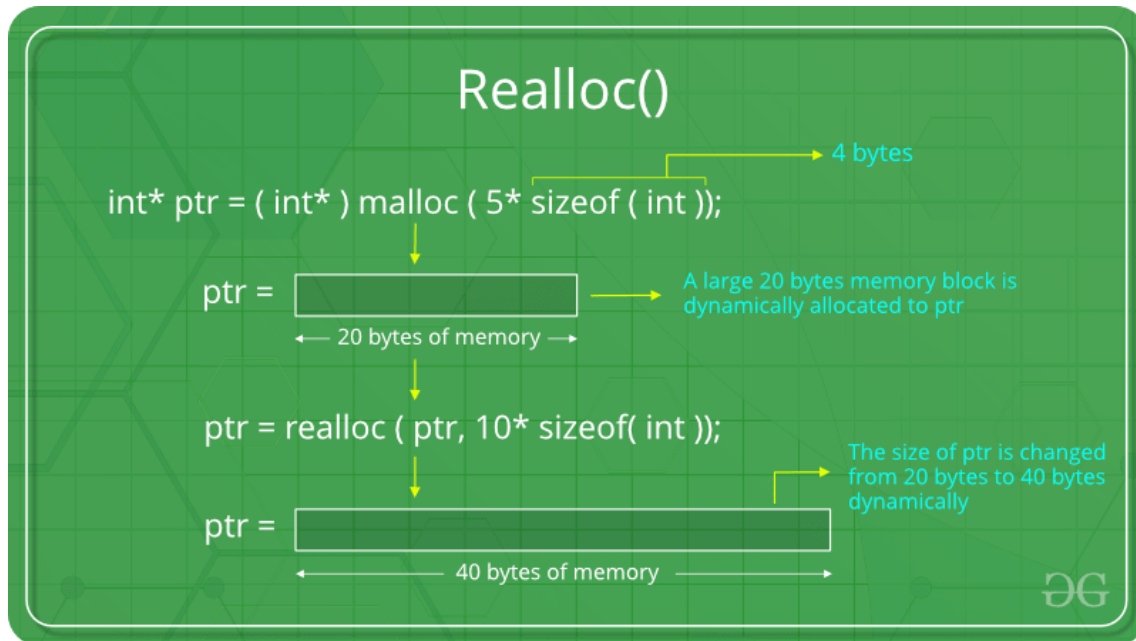
Calloc()

“Contiguous allocation” It’s used to dynamically allocate the specified number of blocks of memory of the specified type. It initializes each block with a default value of 0. If space is insufficient, allocation fails and returns a NULL pointer.



Realloc()

“Re-allocation” If the memory previously allocated with the help of malloc() or calloc() is insufficient, realloc() can be used to dynamically re-allocate memory. re-allocation of memory maintains the already present value and new blocks will be initialized with default garbage value. If space is insufficient, allocation fails and returns a NULL pointer.



Free()

“Free (memory)” It helps to reduce wastage of memory by freeing it. The memory allocated using functions malloc() and calloc() is not de-allocated on its own. The free() method is used whenever the dynamic memory allocation takes place.

