Dynamic Memory Allocation

C//

- Stack destroys on function return

data in Heap is shared through dynamic libraries.

- Run time memory allocation
  ↓
  Decision at runtime
  ↓
  Wastage of memory is restricted

Why dynamic MA in C.

⇒ Changing the size of an array is not allowed.

→ Memory ~~is~~ can be allocation ~~& can be~~ or free at run time.

function for DMA in C

1) Malloc () - Most used, Memory allocation.
2) Calloc ()
3) free ()
4) Realloc ()

1) Malloc ()

It takes number of bytes to be allocated as an input and returns a pointer of type void.

int *ptr.

Syntax : $\text{ptr} = \underbrace{(\text{int}^*)}_{\text{Type cast}}\ \text{malloc}\,(\underbrace{30}_{\text{(space of ints}} * \underbrace{\text{sizeof (int)}}_{(4)}))$

<stdlib.h> header file.

- It can be use as array. (ptr[i])

2) Calloc :- Continous allocation.
It intialize every memory block with a default value of 0.

Syntax : ptr = (float*) calloc (30, sizeof (float));
⇓
Allocates contigure space in memory for 30 blocks floats, with intialize by 0 at starting.

If something goes wrong during memory allocation, then a NULL pointer is returned.

3) free function :- To delocate memory.

The memory allocated by malloc & calloc is not deallocate automatically

4) realloc() function :- To allocate memory of new size using the previous pointer and size.

Syntax :-

ptr = realloc (ptr , newsize)j