Dynamic Memory Allocation

The process of allocating the memory at the time of execution.

Built-in functions declared in the header file <stdlib.h> -

* Permanent Storage Area: Program instructions, Global and Static variables are stored
* Stack: Local variables are stored on the stack
* Heap: Memory space available between permanent storage area and stack can be used for dynamic allocation during program execution and the Size of the heap keeps changing when a program is executed because of the creation and deletion of the variables that are local to the functions and blocks

1. malloc() [Memory Allocation]

Used to dynamically allocate a single large block of contiguous memory according to the size specified.

SYNTAX:   
(void \*)malloc(size);

malloc function simply allocates a memory block according to the size

specified in the heap and on the success it returns a pointer pointing to

the first byte of the allocated memory else returns NULL.

The void pointer can be typecasted to an appropriate type.

Example, int \*ptr = (int \*)malloc(4)

1. calloc() [Clear allocation]

Used to dynamically allocate multiple blocks of memory.

Eg. int \*ptr = (int \*)calloc( 10 , sizeof(int) )

Two Differences from malloc -

A. SYNTAX:

(void \*)calloc(n, size);

{ n - no. of blocks, size - size of each block }

B.Memory allocated to calloc is initialised to 0.  
(Memory allocated to malloc is initialized to some garbage values)

1. free()

Used to release the dynamically allocated memory in heap.

SYNTAX: free(ptr);

The memory allocated in the heap will not be released automatically after using the memory. The space remains there and cannot be used.

1. realloc()

Used to change the size of the memory block without losing the old data.

SYNTAX:

(void \*)realloc(void \*ptr, newSize);

{ void \*ptr - preallocated memory pointer, newSize - new size to be allocated}

Example,

int \*ptr = (int \*)malloc(sizeof(int));

ptr = (int \*)realloc(ptr, 2\*sizeof(int));

Note. we may lose the data when the new size is lower than the old size.