Dynamic Memory Allocation

It is a way to allocate memory to a data structure during the runtime. With help of some functions we can allocate and free memory dynamically.

Functions for Dynamic Memory Allocation

1. malloc() - The malloc or memory allocation function in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form. It doesn't initialize memory at execution time so that it has initialized each block with the default garbage value initially.

   ptr = (*int) malloc(5 *sizeof(int))

2. calloc() - calloc or contiguous allocation function in C is used to dynamically allocate the specified number of blocks of memory of the specified type. it is very much similar to malloc() but has two different points and these are:
   - It initializes each block with a default value '0'.
   - It has two parameters or arguments as compare to malloc().

   ptr = (*int) calloc(5, sizeof(int))

3. free() - free function in C is used to dynamically de-allocate the memory. The memory allocated using functions malloc() and calloc() is not de-allocated on their own. Hence the free() function is used, whenever the dynamic memory allocation takes place. It helps to reduce wastage of memory by freeing it.

   free(ptr);

4. realloc() - realloc or “re-allocation” function in C is used to dynamically change the memory allocation of a previously allocated memory. In other words, 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 the default garbage value.

   ```
   ptr = realloc(ptr, newSize);
   ```
   where ptr is reallocated with new size 'newSize'.