Memory allocation functions – malloc, calloc

Malloc

Memory allocation functions

There are 4 library functions provided by C defined under <stdlib.h> header file to facilitate dynamic memory allocation in C programming. They are:

- malloc()
- calloc()
- free()
- realloc()

malloc():

“malloc” or “memory allocation” method 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.

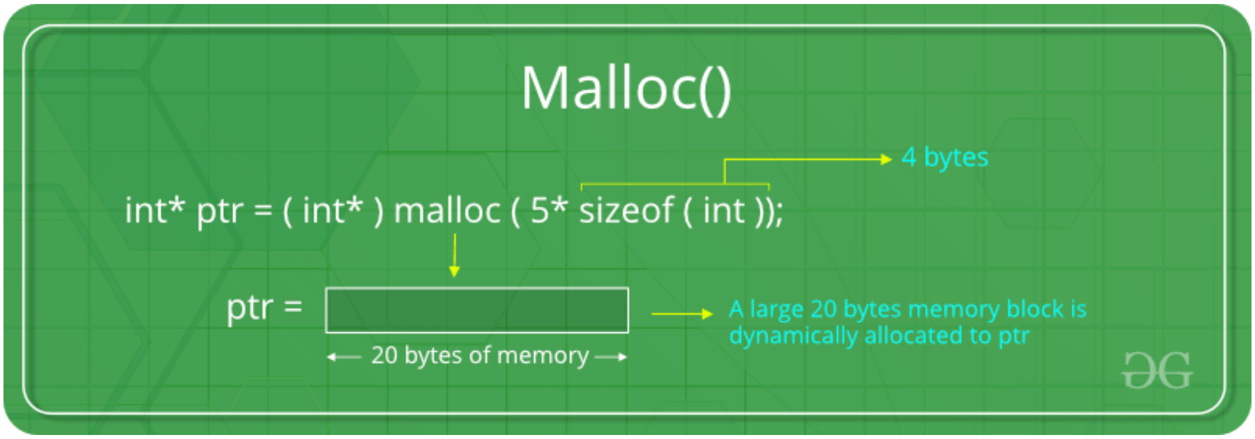
Syntax:

```
ptr = (cast-type*) malloc(byte-size)
```

For Example:

```
ptr = (int*) malloc(100 * sizeof(int));
```

Since the size of int is 4 bytes, this statement will allocate 400 bytes of memory. And, the pointer ptr holds the address of the first byte in the allocated memory.


Malloc()
4 bytes
int* ptr = (int*) malloc (5* sizeof (int));
ptr =
20 bytes of memory
A large 20 bytes memory block is dynamically allocated to ptr

malloc(): example

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
            int* ptr;
            int n=5, i, sum = 0;
            ptr = (int*)malloc(n * sizeof(int));
    for (i = 0; i < n; ++i) {
        ptr[i] = i + 1;
    }
    printf("The elements of the array are: ");
    for (i = 0; i < n; ++i) {
        printf("%d, ", ptr[i]);
    }
            return 0;
}
```

This is DC for C