**C malloc() method**

**“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.

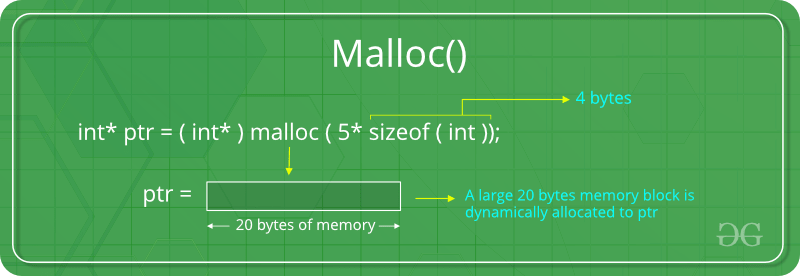
**Syntax:**

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.*



#include <stdio.h>

#include <stdlib.h>

**int** main()

{

    // This pointer will hold the

    // base address of the block created

**int**\* ptr;

**int** n, i, sum = 0;

    // Get the number of elements for the array

    n = 5;

**printf**("Enter number of elements: %d\n", n);

    // Dynamically allocate memory using malloc()

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

    // Check if the memory has been successfully

    // allocated by malloc or not

**if** (ptr == NULL) {

**printf**("Memory not allocated.\n");

**exit**(0);

    }

**else** {

        // Memory has been successfully allocated

**printf**("Memory successfully allocated using malloc.\n");

        // Get the elements of the array

**for** (i = 0; i < n; ++i) {

            ptr[i] = i + 1;

        }

        // Print the elements of the array

**printf**("The elements of the array are: ");

**for** (i = 0; i < n; ++i) {

**printf**("%d, ", ptr[i]);

        }

    }

**return** 0;

}

Enter number of elements: 5

Memory successfully allocated using malloc.

The elements of the array are: 1, 2, 3, 4, 5,

**Output:**

Enter number of elements: 5

Memory successfully allocated using malloc.

The elements of the array are: 1, 2, 3, 4, 5,