

Dynamic Memory Allocation

C is a language with some fixed rules of programming
for example changing the size of an array is not allowed

Dynamic Memory Allocation

Dynamic memory allocation is a way to allocate memory to a data structure during the runtime we can use DMA functions available in C to allocate and free memory during runtime

functions of DMA in C

following functions are available in C to perform Dynamic memory allocation

- (1) malloc()
- (2) calloc()
- (3) free()
- (4) realloc()

malloc () function

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

Syntax:-

$\text{ptr} = (\text{int}^*) \text{malloc} (30 * \text{size of } (\text{int}))$

Space for 30 ints returns size of int

→ Casting void pointer to int

The expression returns a null pointer if the memory cannot be allocated.

Quick Quiz :- write a program to create a dynamic array of 5 floats using malloc().

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    float n = 5;
    float *ptr;
    ptr = (float *) malloc (n * sizeof (float));
    ptr[0] = 3.345;
    ptr[1] = 16.345;
    ptr[2] = 6.345;
    ptr[3] = 56.345;
    ptr[4] = 66.345;
    printf ("%f\n", ptr[0]);
    printf ("%f\n", ptr[1]);
    printf ("%f\n", ptr[4]);
    return 0;
}
```

alloc() function

alloc Syntax Stand for Continuous allocation It initializes each block with a default value of 0.

Syntax

$\text{ptr} = (\text{float}^*) \text{calloc}(30, \text{size of } (\text{float}))$;

↓
Allocates contiguous space in memory
for 30 blocks (floats)

If the space is not sufficient, memory allocation fails and a NULL pointer is returned

Quick Quiz:- write a program to create an array of size n using calloc where n is an integer entered by the user.

3) $\#include <\text{stdio.h}>$
 $\#include <\text{stdlib.h}>$

```
int main() {
    int n;
    int *ptr;
    scanf("%d", &n);
    ptr = (int *)calloc(n, sizeof(int));
    // int arr[n]; // not allowed in C
    ptr[0] = 3;
    printf("%d", ptr[0]);
    return 0;
}
```

free() function

we can use free() function to de allocate the memory

The memory allocated using calloc/malloc is not deallocated automatically

Syntax:

free(ptr); → Memory of ptr is released

Quick Quiz :- write a program to demonstrate the usage of free() with malloc

→ #include < stdio.h>
#include < stdlib.h>

```
int main (){  
    int n;  
    int *ptr;  
    scanf ("%d", &n);  
    ptr = (int *) malloc (n * sizeof (int));  
    ptr[0] = 3;  
    free (ptr);  
    printf ("%d", ptr[0]);  
    return 0;  
}
```

malloc() function

Sometimes the dynamically allocated memory is insufficient or more than required

realloc is used to allocate memory of new size using the previous pointer by size

Syntax:

`ptr = realloc (ptr, newsize);`

`ptr = realloc (ptr, 3 * sizeof (int));`

ptr now points to this
new block of memory capable
of storing 3 integers.