

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 ~~the~~ 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

malloc stands for memory allocation. It takes number of bytes to be allocated as an input and returns a pointer of type void.

Syntax:-

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

(*) Casting void pointer to int

Space for 30 ints

returns size of 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().

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main () {
4     float n = 5;
5     float * ptr;
6     ptr = (float *) malloc (n * sizeof (float));
```

```
7     ptr[0] = 3.345;
8     ptr[1] = 16.345;
9     ptr[2] = 6.345;
10    ptr[3] = 56.345;
11    ptr[4] = 66.345;
```

```
12    printf ("%f\n", ptr[0]);
13    printf ("%f\n", ptr[1]);
14    printf ("%f\n", ptr[4]);
15    return 0;
16 }
```

calloc () function

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

Syntax

```
ptr = (float*) calloc (30, size of (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.

```
#include <stdio.h>
#include <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;
}
```

`realloc()` 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 of 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.