



**Department of Computer Science and Information
Technology**

LAB MANUAL

Data Structures and Algorithm
(III Semester Bsc. CSIT)

Nepathya College
Tilottama-2, Rupandehi

INDEX

1. Introduction to Dynamic Memory Allocation(DMA)

Nepathya College

Ex. no.: 1**Date :****Introduction to Dynamic Memory Allocation(DMA)****Dynamic Memory Allocation:**

The concept of dynamic memory allocation in c language enables the C programmer to allocate memory at runtime. The process of allocating and freeing memory at run time is known as Dynamic Memory Allocation. This reserves the memory required by the program and returns this resource to the system once the use of reserved space is utilized.

Dynamic memory allocation in c language is possible by 4 functions of stdlib.h header file.

malloc()	allocates single block of requested memory.
calloc()	allocates multiple block of requested memory.
realloc()	reallocates the memory occupied by malloc() or calloc() functions.
free()	frees the dynamically allocated memory.

Static memory allocation vs dynamic memory allocation.

static memory allocation	dynamic memory allocation
memory is allocated at compile time.	memory is allocated at run time.
memory can't be increased while executing program.	memory can be increased while executing program.
used in array.	used in linked list.

1. malloc() function in C

The malloc() function allocates single block of requested memory.

It doesn't initialize memory at execution time, so it has garbage value initially.

It returns NULL if memory is not sufficient.

The syntax of malloc() function is given below:

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

Example 1: A program to demonstrate use of malloc() function.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int n,i,*ptr,sum=0;
    printf("Enter number of elements: ");
    scanf("%d",&n);
    ptr=(int*)malloc(n*sizeof(int)); //memory allocated using malloc
    if(ptr==NULL)
    {
        printf("Sorry! unable to allocate memory");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i=0;i<n;++i)
    {
        scanf("%d",ptr+i);
        sum+=*(ptr+i);
    }
    printf("Sum=%d",sum);
    free(ptr);
    return 0;
}
```

Output:

2. calloc() function in C

The calloc() function allocates multiple block of requested memory.

It initially initialize all bytes to zero.

It returns NULL if memory is not sufficient.

The syntax of calloc() function is given below:

```
ptr=(cast-type*)calloc(number, byte-size)
```

Example 2: A program to demonstrate use of calloc() function.

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int n,i,*ptr,sum=0;
    printf("Enter number of elements: ");
    scanf("%d",&n);
    ptr=(int*)calloc(n,sizeof(int)); //memory allocated using calloc
    if(ptr==NULL)
    {
        printf("Sorry! unable to allocate memory");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i=0;i<n;++i)
    {
        scanf("%d",ptr+i);
        sum+=*(ptr+i);
    }
    printf("Sum=%d",sum);
    free(ptr);
    return 0;
```

```
}
```

Output:

3. realloc() function in C

If memory is not sufficient for malloc() or calloc(), you can reallocate the memory by realloc() function. In short, it changes the memory size.

The syntax of realloc() function.

```
ptr=realloc(ptr, new-size)
```

4. free() function in C

The memory occupied by malloc() or calloc() functions must be released by calling free() function. Otherwise, it will consume memory until program exit.

The syntax of free() function.

```
free(ptr)
```

Example 3: A program to demonstrate use of realloc() function.

```
#include <stdio.h>
int main () {
    char *ptr;
    ptr = (char *) malloc(10);
    strcpy(ptr, "Programming");
    printf(" %s, Address = %u\n", ptr, ptr);

    ptr = (char *) realloc(ptr, 20); //ptr is reallocated with new size
    strcat(ptr, " In 'C'");
    printf(" %s, Address = %u\n", ptr, ptr);
    free(ptr);
    return 0;
}
```

Result Analysis:

- We can dynamically manage memory by creating memory blocks as needed in the heap
- In C Dynamic Memory Allocation, memory is allocated at a run time.
- Dynamic memory allocation permits to manipulate strings and arrays whose size is flexible and can be changed anytime in your program.
- It is required when you have no idea how much memory a particular structure is going to occupy.
- Malloc() in C is a dynamic memory allocation function which stands for memory allocation that blocks of memory with the specific size initialized to a garbage value
- Calloc() in C is a contiguous memory allocation function that allocates multiple memory blocks at a time initialized to 0
- Realloc() in C is used to reallocate memory according to the specified size.
- Free() function is used to clear the dynamically allocated memory.