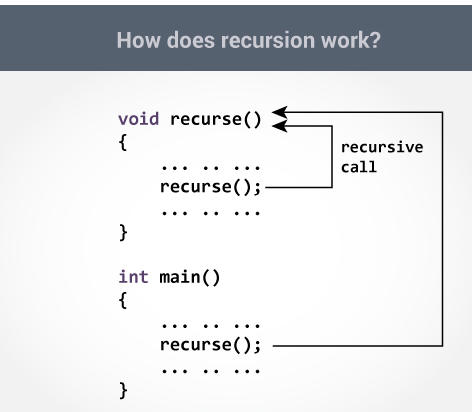
|  |  |
| --- | --- |
| **CS118 Programming Fundamentals** | **LAB 07** Recursion & Arrays |
| **NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES** | |

**Recursion**   
A function that calls itself is known as a recursive function. And, this technique is known as recursion.  


The recursion continues until some condition is met to prevent it. To prevent infinite recursion, [if...else statement](https://www.programiz.com/c-programming/c-if-else-statement) (or similar approach) can be used where one branch makes the recursive call, and other doesn't.

**Example: Sum of Natural Numbers Using Recursion**

#include <stdio.h>

int sum(int n);

int main()

{

int number, result;

printf("Enter a positive integer: ");

scanf("%d", &number);

result = sum(number);

printf("sum = %d", result);

return 0;

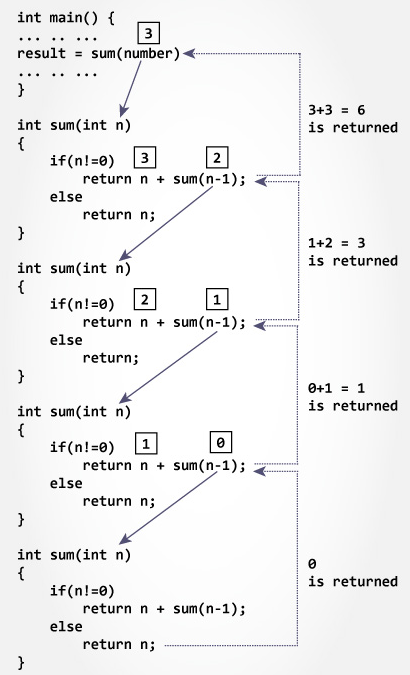
}int sum(int num)

{if (num!=0)

return num + sum(num-1); // sum() function calls itself

else

return num;}



# **ARRAY**

An array is a collection of a fixed number of values of a single type. In C

language, arrays are referred to as structured data types. An array is defined as **finite ordered collection of homogenous** data, stored in contiguous memory locations.

* **Finite** means data range must be defined.
* **Ordered** means data must be stored in continuous memory addresses.
* **Homogenous** means data must be of similar data type.

# Why we need Array in C Programming?

Consider a scenario where you need to find out the average of 100 integer numbers entered by user. In C, you have two ways to do this: 1) Define 100 variables with int data type and then perform 100 scanf() operations to store the entered values in the variables and then at last calculate the average of them. 2) have a single integer array to store all the values, loop the array to store all the entered values in array and later calculate the average.

## Which solution is better according to you? AND WHY?

Arrays are of two types:

1. One-dimensional arrays
2. [Multidimensional arrays](https://www.programiz.com/c-programming/c-multi-dimensional-arrays)

# ONE DIMENSIONAL ARRAY

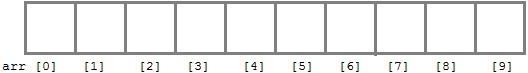
**Declaring an Array**

Like any other variable, arrays must be declared before they are used. General form of array declaration is,

data-type variable-name[size];

/\* Example of array declaration \*/

int arr[10];



Index of an array starts from *0* to size-1 i.e first element of *arr* array will be stored at arr[0]address and the last element will occupy arr[9]

**Compile time Array initialization**

Compile time initialization of array elements is same as ordinary variable initialization. The general form of initialization of array is,

data-type array-name[size] = { list of values };

/\* Here are a few examples \*/

int marks[4]={ 67, 87, 56, 77 }; // integer array initialization

float area[5]={ 23.4, 6.8, 5.5 }; // float array initialization

int marks[4]={ 67, 87, 56, 77, 59 }; // Compile time error

One important thing to remember is that when you will give more initializer (array elements) than the declared array size than the compiler will give an error.

#include<stdio.h>

void main()

{

int i;

int arr[] = {2, 3, 4}; // Compile time array initialization

for(i = 0 ; i < 3 ; i++)

{

printf("%d\t",arr[i]);

}}

**Output:**

**2 3 4**

**Runtime Array initialization**

An array can also be initialized at runtime using scanf() function. This approach is usually used for initializing large arrays, or to initialize arrays with user specified values. Example,

#include<stdio.h>

void main()

{

int arr[4];

int i, j;

printf("Enter array element");

for(i = 0; i < 4; i++)

{

scanf("%d", &arr[i]); //Run time array initialization

}

for(j = 0; j < 4; j++)

{

printf("%d\n", arr[j]);

}

}

**Lab Task**

**Question # 01:**

(a) Write a program to find Factorial of a Number Using Recursion

(b)Write a program to count the digits of a given number using recursion function.

**Question # 02:**

Write a program to GCD of two numbers using recursion function.

HINT: The GCD of 24 and 60 is 2 × 2 × 3 = 12.

**Question # 03:**

Write a program to Print Fibonacci Series using recursion function.

**Question # 04:**

Write a C program to store 10 decimal numbers in an array and display it on output screen. Also find the sum of all 10 numbers.

**Question # 05:**

Suppose your Programming Fundamental teacher provided you the midterm marks of your section. Your task is to find the highest marks and 2nd highest marks from the given list by C programming. Also find the sum of total marks of all the students and average of the marks

2d Array Sorting

Recursion