

**এখন আমরা দুটি সংখ্যার গসাণ্ড (GCD → Greatest Common Divisor বা HCF → Highest Common Factor) ও লসাণ্ড (LCM → Least Common Multiple) নির্ণয় করার জন্য প্রোগ্রাম লিখব।**

দুটি সংখ্যার গসাণ্ড হচ্ছে যেসব সংখ্যা দিয়ে ওই দুটি সংখ্যা নিঃশেষে বিভাজ্য হয়, তাদের মধ্যে সবচেয়ে বড় সংখ্যা। তাহলে আমরা যেটি করব, দুটি সংখ্যা  $a$  ও  $b$  নেব। তারপর এদের মধ্যে যেটি ছোট, সেই মানটি আবার  $x$  ভেরিয়েবলে রাখব। গসাণ্ড এর মান  $x$ -এর চেয়ে বড় হওয়া সম্ভব নয় (5 ও 10-এর গসাণ্ড-এর মান নিশ্চয়ই 5-এর চেয়ে বড় হবে না)। এখন  $a$  ও  $b$ ,  $x$  দিয়ে নিঃশেষে বিভাজ্য হয় কি না ( $a \% x == 0$  এবং  $b \% x == 0$ ) সেটি পরীক্ষা করব। যদি হয় তবে আমরা গসাণ্ড পেয়ে গেছি। যদি  $a$  ও  $b$  উভয়েই নিঃশেষে বিভাজ্য না হয়, তখন  $x$ -এর মান এক কমিয়ে পরীক্ষা করব। যতক্ষণ না আমরা গসাণ্ড পাচ্ছি  $x$ -এর মান কমাতেই থাকব। একসময় আমরা গসাণ্ড পাবই, কারণ  $x$ -এর মান যখন 1 হবে, তখন তো  $x$  দিয়ে  $a$  ও  $b$  দুটি সংখ্যাই নিঃশেষে বিভাজ্য। তোমরা কি প্রোগ্রামটি নিজে লিখার চেষ্টা করবে? না পারলে আমার কোড দেখো:

```
#include <stdio.h>
int main()
{
    int a, b, x, gcd;
    scanf("%d %d", &a, &b);
    if (a < b) {
```

```

    x = a;
}
else {
    x = b;
}
for(; x >= 1; x--) {
    if (a % x == 0 && b % x == 0) {
        gcd = x;
        break;
    }
}
printf("GCD is %d\n", gcd);
return 0;
}

```

প্রোগ্রামে দেখো gcd পাওয়ার সঙ্গে সঙ্গে লুপ থেকে বের হয়ে যেতে হবে (আমি break ব্যবহার করেছি এই জন্য)। break ব্যবহার না করলে কী হবে সেটি পরীক্ষা করে দেখো।

তবে গসাগু বের করার জন্য আমি যেই পদ্ধতি ব্যবহার করেছি সেটি খুব সহজ পদ্ধতি হলেও ইফিশিয়েন্ট (efficient) নয়। যেমন, সংখ্যা দুটি খুব বড় হলে এবং সহমৌলিক (co-prime) হলে লুপটি কিন্তু অনেকবার ঘুরবে। কারণ সহমৌলিক হলে গসাগু হবে 1।

**তোমরা নিশ্চয়ই জানো যে, দুটি সংখ্যার মধ্যে 1 ছাড়া আর কোনো সাধারণ উৎপাদক না থাকলে সংখ্যা দুটি সহমৌলিক।**

গসাণ্ড বের করার জন্য ইউক্লিডের একটি চমৎকার পদ্ধতি আছে। ইউক্লিড ভাগশেষ উপপাদ্যের (division algorithm) সাহায্যে গসাণ্ড বের করার উপায় দেখিয়েছেন। এই পদ্ধতিতে খুব সহজে গসাণ্ড বের করা যায় এবং প্রোগ্রামটিও বেশ ইফিশিয়েন্ট হয়। এর জন্য দুটি জিনিস জানা লাগবে:

$a$  ও  $0$ -এর গসাণ্ড-এর মান  $a$ ।

$a$  ও  $b$ -এর গসাণ্ড =  $b$  ও  $a \% b$ -এর গসাণ্ড।

তাহলে প্রোগ্রামে যেটি করতে হবে, একটি লুপের সাহায্যে  $a$ -এর মান  $b$  আর  $b$ -এর মান  $a \% b$  বসিয়ে যেতে হবে, যতক্ষণ না  $b$ -এর মান শূন্য হয়।  $b$ -এর মান শূন্য হলেই বুঝে যাব যে গসাণ্ড হচ্ছে  $a$  (এটা কিন্তু প্রোগ্রাম শুরুর সময়  $a$ -এর মান না,  $b$ -এর মান যখন শূন্য হবে সেই সময়  $a$ -এর মান)।

Bismillahhir Rahmanir Rahim

Let's appear in thy smiling.  
~ W. S. Peare



**SOROWAR MAHABUB**

Studies B.Sc in CSE,  
Int' Islamic University Chittagong

Find me on:

Facebook: Sorowar Mahabub

Instagram: Eng.\_Sorowar\_Mahabub

Cell: 01521-564157, 01834-756433

Mail: sorowarmahabub.bsc.cse.iiuc@gmail.com

```
#include <stdio.h>
int main()
{
    int a, b, t, x, gcd;
    scanf("%d %d", &a, &b);
    if (a == 0) gcd = b;
    else if (b == 0) gcd = a;
    else {
        while (b != 0) {
            t = b;
            b = a % b;
            a = t;
        }
        gcd = a;
    }
    printf("GCD is %d\n", gcd);
    return 0;
}
```

এবার লসাগু বের করার প্রোগ্রাম। তোমরা নিশ্চয়ই স্কুলে শিখেছ, কীভাবে লসাগু বের করতে হয়। সেই পদ্ধতি অবলম্বন করে

প্রোগ্রাম লিখে ফেলো। আর যারা সেই পদ্ধতি জানো না, তাদের জন্য একটি সূত্র বলে দিচ্ছি। আশা করি, লসাগু বের করার প্রোগ্রাম লিখতে আর সমস্যা হবে না।

**দুটি সংখ্যার লসাগু x দুটি সংখ্যার গসাগু = সংখ্যা দুটির গুণফল।**

Bismillahir Rahmanir Rahim



Let's appear in thy smiling.  
~ W. S. Peare



## SOROWAR MAHABUB

Studies B.Sc in CSE,  
Int' Islamic University Chittagong

Find me on:  
Facebook: Sorowar Mahabub  
Instagram: Eng.\_Sorowar\_Mahabub  
Cell: 01521-564157, 01834-756433  
Mail: sorowarmahabub.bsc.cse.iiuc@gmail.com

# Type Casting

Converting one datatype into another is known as type casting or, type-conversion. For example, if you want to store a 'long' value into a simple integer then you can type cast 'long' to 'int'. You can convert the values from one type to another explicitly using the cast operator as follows –

# (type\_name) expression

Consider the following example where the cast operator causes the division of one integer variable by another to be performed as a floating-point operation –

```
#include <stdio.h>

main() {

    int sum = 17, count = 5;
    double mean;

    mean = (double) sum / count;
    printf("Value of mean : %f\n", mean );
}
```

When the above code is compiled and executed, it produces the following result –

Value of mean : 3.400000

It should be noted here that the cast operator has precedence over division, so the value of sum is first converted to type double and finally it gets divided by count yielding a double value.

Bismillahir Rahmanir Rahim

Let's appear in thy smiling.  
~ W. S. Peare



**SOROWAR MAHABUB**

Studies B.Sc in CSE,  
Int' Islamic University Chittagong

---

Find me on:  
Facebook: Sorowar Mahabub  
Instagram: Eng.\_Sorowar\_Mahabub  
Cell: 01521-564157, 01834-756433  
Mail: sorowarmahabub.bsc.cse.iiuc@gmail.com

# Arrays in C

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.

Here the words,

**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.

*Example* where arrays are used,  
to store list of Employee or Student names,  
to store marks of students,  
or to store list of numbers or characters etc.

Since arrays provide an easy way to represent data, it is classified amongst the data structures in C. Other data structures in c are structure, lists, queues, trees etc. Array can be used to represent not only simple list of data but also table of data in two or three dimensions.

## 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];**



Here int is the data type, arr is the name of the array and 10 is the size of array. It means array arr can only contain 10 elements of int type.

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].

### Initialization of an Array

After an array is declared it must be initialized. Otherwise, it will contain garbage value(any random value). An array can be initialized at either compile time or at runtime.

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]);
    }
}
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]);  
    }  
}
```

Bismillahhir Rahmanir Rahim

Let's appear in thy smiling.  
~ W. S. Peare



# SOROWAR MAHABUB

Studies B.Sc in CSE,  
Int' Islamic University Chittagong

Find me on:

Facebook: Sorowar Mahabub

Instagram: Eng.\_Sorowar\_Mahabub

Cell: 01521-564157, 01834-756433

Mail: sorowarmahabub.bsc.cse.iiuc@gmail.com

# Two dimensional Arrays

C language supports multidimensional arrays also. The simplest form of a multidimensional array is the two-dimensional array. Both the row's and column's index begins from 0.

Two-dimensional arrays are declared as follows,

```
data-type array-name[row-size][column-size]
```

`/* Example */`

```
int a[3][4];
```

An array can also be declared and initialized together. For example,

```
int arr[][3] =  
{  
    {0,0,0},  
    {1,1,1}  
};
```

Note: We have not assigned any row value to our array in the above example. It means we can initialize any number of rows. But, we must always specify number of columns,

else it will give a compile time error. **Here, a 2\*3 multi-dimensional matrix is created.**

Runtime initialization of a two dimensional Array

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

```
void main()
```

```
{  
    int arr[3][4];  
    int i, j, k;  
    printf("Enter array element");  
    for(i = 0; i < 3;i++)  
    {  
        for(j = 0; j < 4; j++)  
        {  
            scanf("%d", &arr[i][j]);  
        }  
    }  
    for(i = 0; i < 3; i++)  
    {  
        for(j = 0; j < 4; j++)  
        {  
            printf("%d", arr[i][j]);  
        }  
    }  
}
```

# Arrays

## Overview

An array is a collection of data items, all of the same type, accessed using a common name.

A one-dimensional array is like a list; A two dimensional array is like a table; The C language places no limits on the number of dimensions in an array, though specific implementations may.

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? Obviously the second solution, it is convenient to store same data types in one single variable and later access them using array index (we will discuss that later in this tutorial).

## How to declare Array in C

```
int num[35]; /* An integer array of 35 elements */
char ch[10]; /* An array of characters for 10 elements */
```

Similarly an array can be of any data type such as double, float, short etc.

## How to access element of an array in C

You can use array subscript (or index) to access any element stored in array. Subscript starts with 0, which means `arr[0]` represents the first element in the array `arr`.

In general `arr[n-1]` can be used to access `n`th element of an array. where `n` is any integer number.

For example:

```
int mydata[20];
mydata[0] /* first element of array mydata*/
mydata[19] /* last (20th) element of array mydata*/
```

Example of Array In C programming to find out the average of 4 integers

```
#include <stdio.h>
int main()
{
    int avg = 0;
    int sum =0;
    int x=0;
```



```
/* Array- declaration – length 4*/
int num[4];

/* We are using a for loop to traverse through the array
 * while storing the entered values in the array
 */
for (x=0; x<4;x++)
{
    printf("Enter number %d \n", (x+1));
    scanf("%d", &num[x]);
}
for (x=0; x<4;x++)
{
    sum = sum+num[x];
}

avg = sum/4;
printf("Average of entered number is: %d", avg);
return 0;
}
```

Output:

```
Enter number 1
10
Enter number 2
10
Enter number 3
20
```

Enter number 4

40

Average of entered number is: 20

Lets discuss the important parts of the above program:

Input data into the array

Here we are iterating the array from 0 to 3 because the size of the array is 4. Inside the loop we are displaying a message to the user to enter the values. All the input values are stored in the corresponding array elements using scanf function.

```
for (x=0; x<4;x++)  
{  
    printf("Enter number %d \n", (x+1));  
    scanf("%d", &num[x]);  
}
```

Reading out data from an array

Suppose, if we want to display the elements of the array then we can use the for loop in C like this.

```
for (x=0; x<4;x++)  
{  
    printf("num[%d]\n", num[x]);  
}
```

Various ways to initialize an array

In the above example, we have just declared the array and later we initialized it with the values input by user. However you can also initialize the array during declaration like this:

```
int arr[5] = {1, 2, 3, 4 ,5};
```

OR (both are same)

```
int arr[] = {1, 2, 3, 4, 5};
```

Un-initialized array always contain garbage values.

Bismillahir Rahmanir Rahim

Let's appear in thy smiling.  
~ W. S. Peare



# SOROWAR MAHABUB

Studies B.Sc in CSE,  
Int' Islamic University Chittagong

---

Find me on:  
Facebook: Sorowar Mahabub  
Instagram: Eng.\_Sorowar\_Mahabub  
Cell: 01521-564157, 01834-756433  
Mail: sorowarmahabub.bsc.cse.iiuc@gmail.com