



**Subject: C Programming**

**Sem: I**

✓ **4.1 Arrays**

**Concept**

- An array stores multiple values of the same data type in contiguous memory.
- Index starts from 0.

**Declaration & Definition**

```
int arr[5];      // Declaration (size known)
int arr[5] = {1,2,3,4,5}; // Declaration + Initialization
```

**Accessing Array Elements**

```
printf("%d", arr[0]); // Access 1st element
arr[2] = 10;          // Modify element at index 2
```

**One-Dimensional Array Example**

```
int marks[3] = {85, 90, 78};

for(int i=0; i<3; i++) {
    printf("%d ", marks[i]);
}
```

**Multidimensional Array (2D Array)**

**Declaration**

```
int a[2][3]; // 2 rows, 3 columns
```

**Initialization**

```
int a[2][3] = {
    {1,2,3},
    {4,5,6}
};
```

**Access Example**

```
printf("%d", a[1][2]); // Output: 6
```

✓ **4.2 Strings**

**Basics of String**

- A string is a character array ending with a null character \0.

**Example**

```
char name[] = "Ashfaque";
```

**Array of Strings**

```
char fruits[3][10] = {"Apple", "Mango", "Orange"};
```

```
printf("%s", fruits[1]); // Output: Mango
```

### Functions in <string.h>

Function	Purpose	Example
<b>strlen()</b>	Length of string	strlen(name)
<b>strcpy()</b>	Copy string	strcpy(dest, src);
<b>strcat()</b>	Concatenate strings	strcat(s1, s2);
<b>strcmp()</b>	Compare strings	strcmp(s1, s2)

### Examples:

```
#include <string.h>

char s1[20] = "Hello";
char s2[] = "World";

int len = strlen(s1);      // 5
strcat(s1, s2);           // s1 = HelloWorld
strcpy(s1, s2);           // s1 = World
int cmp = strcmp("A", "B"); // negative value
```

## 4.3 Structures

### Declaration

```
struct Student {
    char name[20];
    int age;
    float marks;
};
```

### Initialization

```
struct Student s1 = {"Ashfaque", 25, 88.5};
```

### Accessing Members

```
printf("%s", s1.name);
s1.age = 26; // Modify
```

### Structure within Structure

```
struct Address {
    char city[20];
    int pincode;
};

struct Student {
    char name[20];
    struct Address addr;
};

struct Student s = {"Ashfaque", {"Mumbai", 400001}};
printf("%s", s.addr.city);
```

### Operations on Structures

- ✓ Assign
- ✓ Copy
- ✓ Compare (using functions, not ==)
- ✓ Pass to functions

### Example

```
struct A { int x; };
struct A a1 = {10}, a2;
a2 = a1; // Copy
```

### Array of Structures

```
struct Student {
    char name[20];
    int age;
};

struct Student s[2] = {
    {"Ali", 20},
    {"Rehan", 22}
};

printf("%s", s[1].name); // Output: Rehan
```

### Problem Statement

Write a C program to store 5 integers in an array and find the **largest number**.

### Solution

```
#include <stdio.h>

int main() {
    int arr[5], i, max;

    printf("Enter 5 numbers:\n");
    for(i = 0; i < 5; i++) {
        scanf("%d", &arr[i]);
    }

    max = arr[0]; // assume first element is max

    for(i = 1; i < 5; i++) {
        if(arr[i] > max)
            max = arr[i];
    }

    printf("Largest number = %d", max);

    return 0;
}
```

## 2. STRING – Problem & Solution

### Problem Statement

Write a C program to input a string and count how many **vowels** it contains.

### Solution

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

int main() {
    char str[50];
    int i, count = 0;

    printf("Enter a string: ");
    gets(str);

    for(i = 0; str[i] != '\0'; i++) {
        char ch = str[i];

        if(ch=='a'|| ch=='e'|| ch=='i'|| ch=='o'|| ch=='u' ||
           ch=='A'|| ch=='E'|| ch=='I'|| ch=='O'|| ch=='U')
        {
            count++;
        }
    }

    printf("Number of vowels = %d", count);

    return 0;
}
```

Mohd Ashfaque Shaikh (AIML)

### 3. STRUCTURE – Problem & Solution

#### Problem Statement

Define a structure **Student** with name, roll number, and marks.

Write a program to input data of **3 students** and display the **student with the highest marks**.

#### Solution

```
#include <stdio.h>

struct Student {
    char name[20];
    int roll;
    float marks;
};

int main() {
    struct Student s[3];
    int i, topIndex = 0;

    for(i = 0; i < 3; i++) {
        printf("Enter name, roll, marks of student %d:\n", i+1);
        scanf("%s %d %f", s[i].name, &s[i].roll, &s[i].marks);
    }

    for(i = 1; i < 3; i++) {
        if(s[i].marks > s[topIndex].marks)
            topIndex = i;
    }

    printf("\nTopper:\n");
    printf("Name: %s\n", s[topIndex].name);
    printf("Roll: %d\n", s[topIndex].roll);
    printf("Marks: %.2f\n", s[topIndex].marks);

    return 0;
}
```