

Chapter 8 – Arrays and Strings**✔ Short Questions with Easy Answers****Q1. Define array?**

👉 An array is a **collection of variables of the same type** stored in continuous memory locations.

- Example: `int marks[5];` can store 5 integers.
-

Q2. Define 2D array?

👉 A **2D array (matrix)** is like a table with rows and columns.

- Example: `int matrix[3][3];` // 3x3 matrix
-

Q3. How to declare simple array?

👉 Syntax:

```
type arrayName[size];
```

Example:

```
double balance[10];
```

Q4. How to initialize array?

👉 Arrays can be initialized during declaration:

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

Or without size:

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

Q5. How to store data in 2D array?

👉 For `int arr[5][4];` → total elements = $5 * 4 = 20$.

- First index = row, second index = column.
-

Q6. How to initialize 2D array?

```
int arr[2][4] = {
    {10, 11, 12, 13},
    {14, 15, 16, 17}
};
```

Q7. Define string?

👉 A **string** is a sequence of characters ending with a **null character \0**.

- Example: "Hello" is stored as { 'H','e','l','l','o','\0' }.
-

Q8. Write declaration syntax of string.

👉 `char str_name[size];`

Example: `char name[20];`

Q9. Define strlen function.

👉 `strlen(str)` returns the **length of string** (without `\0`).

Q10. Write syntax of strcat() function.

👉 `char *strcat(char *destination, const char *source);`

- It joins two strings.
-

Long Questions with Easy Answers & Examples

Q1. Explain strcat() function with example.

👉 `strcat()` joins two strings.

Example:

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

int main() {
    char str1[20] = "Hello ";
    char str2[] = "World";
    strcat(str1, str2);
    printf("%s", str1);
    return 0;
}
```

Output: Hello World

Q2. Program: Convert lowercase to uppercase.

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

int main() {
    char str[100];
    printf("Enter a string: ");
    gets(str);

    for (int i = 0; str[i] != '\0'; i++) {
        str[i] = toupper(str[i]);
    }
}
```

```
printf("Uppercase: %s", str);
return 0;
}
```

Q3. Program: Sort names alphabetically.

```
#include <stdio.h>
#include <string.h>
int main() {
    char names[5][20] = {"Ali", "Zara", "Ahmed", "Bilal", "Omar"};
    char temp[20];

    for (int i=0; i<5; i++) {
        for (int j=i+1; j<5; j++) {
            if (strcmp(names[i], names[j]) > 0) {
                strcpy(temp, names[i]);
                strcpy(names[i], names[j]);
                strcpy(names[j], temp);
            }
        }
    }

    printf("Sorted names:\n");
    for (int i=0; i<5; i++)
        printf("%s\n", names[i]);

    return 0;
}
```

Q4. Program: Delete vowels from a sentence.

```
#include <stdio.h>
#include <string.h>
int main() {
    char str[100], result[100];
    int i, j=0;
    printf("Enter a sentence: ");
```

```

gets(str);

for(i=0; str[i]!='\0'; i++) {
    if (!(strchr("AEIOUaeiou", str[i]))) {
        result[j++] = str[i];
    }
}

result[j] = '\0';

printf("Without vowels: %s", result);

return 0;
}

```

Q5. Explain 2D array with example.

👉 2D array = collection of rows & columns.

Example: Store marks of 2 students in 3 subjects:

```

#include <stdio.h>

int main() {
    int marks[2][3] = {
        {85, 90, 80},
        {75, 88, 92}
    };

    for (int i=0; i<2; i++) {
        for (int j=0; j<3; j++) {
            printf("%d ", marks[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

Output:

85 90 80

75 88 92

◆ **Extra Practice Questions with Answers**

Q1. Difference between array and string?

- Array = collection of similar data (e.g., int, float).
 - String = array of **characters** ending with \0.
-

Q2. Write a program to find the largest number in an array.

```
#include <stdio.h>

int main() {

    int arr[5] = {10, 50, 20, 80, 30}, max = arr[0];

    for (int i=1; i<5; i++) {

        if (arr[i] > max)

            max = arr[i];

    }

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

    return 0;

}
```

Q3. Write a program to reverse a string.

```
#include <stdio.h>

#include <string.h>

int main() {

    char str[50];

    printf("Enter string: ");

    gets(str);

    strrev(str); // built-in function

    printf("Reversed: %s", str);

    return 0;

}
```