

Raj Kumar Govt Institute of Technology, Ghazibabad			
Semester Test 1 (CSE Semesters 2020-21)			
Subject Name: PPS			
Subject Code: BCS 101		Total Marks: 50	
Time: 30 min		Time: 30 min	
<p>Section - I</p> <p>Answer all questions. If you require any thing, then show it to the invigilator.</p>			
Q1. a) Present all parts of the following. (2 x 5=10)		K1	C001 - 2
b) Define the following: how do you define a recursive code?		K1	C001 - 2
Q2. Write the features of this C++ code?		K2	C002 - 2
<pre> int main() { int i=5; while(i>0) { cout<<"5\n"; i--; } }</pre>			
Q3. What is linker in C++ language and its proper diagram?		K1	C002 - 2
<p>Section - 2</p>			
Q4. a) Present all parts of the following. (4 x 4=16)			
b) Explain the following: how do you define a recursive code?		K1	C001 - 4
c) Complete a program to calculate the sum of all numbers in an array.		K1	C001 - 4
d) Draw a flow chart and program to calculate the sum of even and odd numbers.		K2	C001 - 4
<p>Section - 3</p>			
Q5. a) Present all parts of the following. (10 x 1=10)			
b) Draw the architecture of digital computer system and explain its components.		K1	C001 - 10
c) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
d) Write a program to calculate the sum of even and odd numbers.		K2	C001 - 10
e) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
f) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
g) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
h) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
i) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10
j) Write a program to calculate the sum of all numbers in an array.		K1	C001 - 10

In C programming, an array is a collection of elements of the same data type, stored in contiguous memory locations. Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

1. **One-Dimensional Array** – A simple list of elements.
2. **Two-Dimensional Array** – Used to represent matrices or tables.
3. **Multi-Dimensional Array** – Arrays with more than two dimensions (rarely used)

Syntax: *(and points)*

Example:

```
#include <stdio.h>
int main()
{
    int marks[5] = {85, 90, 78, 92, 88};
    for(int i = 0; i < 5; i++) {
        printf("marks[%d] = %d\n", i, marks[i]);
    }
    return 0;
}
```

Diagram illustrating memory addresses for the array `marks`:

- Array `marks` is shown as a box divided into 5 cells.
- Addresses are labeled below the cells: `1000`, `1004`, `1008`, `1012`, `1016`.
- A red arrow points from the first cell (address 1000) to the text "Pointer Variable".
- A red arrow points from the second cell (address 1004) to the text "address of a variable".

Output:

```
marks[0] = 85
marks[1] = 90
marks[2] = 78
marks[3] = 92
marks[4] = 88
```

Ans: $arr[] = \{12, 14, 10, 20\}$;

Syntax:

Example

```
#include <stdio.h>
int main() {
```

```
int matrix[2][3] = {{1, 2, 3}, {4, 5, 6}};
for(int i = 0; i < 2; i++) {
    for(int j = 0; j < 3; j++) {
        printf("%d ", matrix[i][j]);
    }
    printf("\n");
}
```

}
return 0;

}

Output:

- Array elements are accessed using their index.

- Indexing starts from 0.
Example: marks[0] → first element, marks[4] → last element

- Array size must be a **constant** at compile time (in standard C)

- The name of the array (e.g., marks) represents the **base address** (starting address) of the array
- You can use arrays with **loops** to perform operations easily.

```
#include <stdio.h>
int main() {
    int arr[5] = {10, 20, 30, 40, 50};
    int sum = 0;
    for(int i = 0; i < 5; i++) {
        sum += arr[i];
    }
    printf("Sum = %d", sum);
    return 0;
}
```

Output:

Sum = 150

```
int a = 4, 10;
```

Localhost (1 byte)

20 byte
200 byte

#include

```
#include <stdio.h>
void inputArray(int arr[], int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("Enter the Value of  
scanf("%d", &arr[i]);
    }
}

void printArray(int arr[], int n)
{
    printf("The Array is:\n");
    printf("\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n\n");
}

int main()
{
    int arr[5];
    inputArray(arr, 5);
    printArray(arr, 5);
    return 0;
}
```

[illegible]

```
int x = 20;
printf("Enter value")
{
    scanf("%d", &x);
    printf("%d", x);
}
2000
X 12
```

```
for (i=0; i<10; i++)
    printf("%d", arr[i]);
```

[illegible]

$n = 5$

0, 1, 2, 3, 4, 5

$$\frac{n(n+1)}{2}$$
$$= \frac{5(5H)}{2} \Rightarrow \frac{5 \times 8}{2}$$
$$15 - 0 = 15 - 2$$
$$= 13 - 3 = 10 - 1 = 9 - 1 = 8$$

76 = 0000

$\frac{96 \times 1000}{100} = 960$

$$\alpha \vee \gamma \beta \quad 4 \mid 5$$

per numoli

1001 ~ 0011

$i=2$

$i = 3$ $x = 0100$

$i=4$

 $x = 1$