

**Functions: MCQs (20 Questions)**

1. Which of the following is a built-in function in C?
  - A. sum()
  - B. printf()
  - C. average()
  - D. compute()
2. A user-defined function must contain:
  - A. Only arguments
  - B. Only return type
  - C. Return type, function name, body
  - D. Header file
3. What is the correct syntax for defining a function?
  - A. return\_type function\_name(parameters);
  - B. return\_type function\_name(parameters) { }
  - C. function\_name return\_type(parameters) { }
  - D. return\_type { function\_name() }
4. A function that does not return a value uses:
  - A. int
  - B. float
  - C. void
  - D. null
5. Passing parameters by value means:
  - A. Address is passed
  - B. Copy of value is passed
  - C. No value is passed
  - D. String is passed only
6. What is local scope?
  - A. Accessible everywhere
  - B. Accessible only inside function/block
  - C. Accessible in all files
  - D. Accessible in OS
7. What is the output?

```
int fun(int x){ x = x + 10; return x; }  
int main(){ int a = 5; fun(a); printf("%d", a); }
```

Programming for Problem Solving  
UNIT – III  
Worksheet – on Functions, Recursion, Pointers

- A. 5
  - B. 10
  - C. 15
  - D. Error
8. Global variables are stored in:
- A. Stack
  - B. Heap
  - C. Code segment
  - D. Data segment
9. A function that calls itself is known as:
- A. Inline function
  - B. Recursive function
  - C. Nested function
  - D. Derived function
10. Which of the following is NOT a category of function?
- A. With arguments, no return
  - B. No arguments, no return
  - C. No arguments, some return, some not
  - D. With arguments, with return
11. Function prototype is written to:
- A. Allocate memory
  - B. Declare function before use
  - C. Increase speed
  - D. Remove errors
12. What is array of functions?
- A. Array storing addresses of functions
  - B. Array storing integers only
  - C. Function returning an array
  - D. None
13. Which keyword is used to return multiple values?
- A. return
  - B. break
  - C. none
  - D. Not possible directly
14. Inline functions are used to:
- A. Reduce memory
  - B. Reduce function call overhead

- C. Increase recursion  
D. Replace main()
15. A function with same name but different parameters is called:  
A. Overloading  
B. Overriding  
C. Nesting  
D. Recursion
16. What is the default return type of a function in C?  
A. void  
B. float  
C. int  
D. char
17. Which is valid?  
A. return(a, b);  
B. return a, b;  
C. return a + b;  
D. return(a;)
18. The variables passed to a function are called:  
A. arguments  
B. parameters  
C. identifiers  
D. members
19. Functions help in:  
A. Reusability  
B. Modularity  
C. Debugging  
D. All of these
20. What is the output?

```
void fun(){ printf("A"); }  
int main(){ fun(); fun(); }
```

- A. A  
B. AA  
C. AAA  
D. None

**Recursion: MCQs (20 Questions)**

1. Recursion works on the principle of:
  - A. FIFO
  - B. LIFO
  - C. Random
  - D. Static
2. A recursive function must have:
  - A. One base case
  - B. One recursive call
  - C. Base case + recursive case
  - D. No condition
3. What is the output?

```
int fun(int n){ if(n==0) return 1; return fun(n-1) + 1; }  
printf("%d", fun(3));
```

- A. 3
  - B. 4
  - C. 1
  - D. 0
4. Stack overflow occurs when:
  - A. Infinite recursion happens
  - B. Memory is free
  - C. File not found
  - D. CPU overheats
5. Factorial recursive function stops when:
  - A.  $n < 0$
  - B.  $n == 1$  or 0
  - C.  $n == 10$
  - D.  $n$  is float
6. What is traced first in recursion?
  - A. Base case
  - B. Recursive case
  - C. Return phase
  - D. Intermediate
7. Fibonacci recursion contains:
  - A. Single recursion

- B. Double recursion
  - C. No recursion
  - D. Infinite loop
8. Recursion with arrays requires:
- A. Passing array name
  - B. Passing size
  - C. Passing index
  - D. All of these

9. What is the output?

```
void fun(int n){ if(n==0) return; printf("%d",n); fun(n-1); }  
fun(3);
```

- A. 123
  - B. 321
  - C. 111
  - D. Error
10. The return phase in recursion happens:
- A. First
  - B. Last
  - C. Middle
  - D. Never
11. What is tail recursion?
- A. Recursion at top
  - B. Recursion at bottom
  - C. Last statement is recursive call
  - D. First statement is recursive call
12. Which is better for very deep calls?

- A. Recursion
- B. Loop
- C. Switch
- D. Array

13. What is printed?

```
int fun(int n){ if(n==1) return 1; return n + fun(n-1); }  
printf("%d", fun(3));
```

- A. 3
- B. 4
- C. 6
- D. 1

14. Recursion is stored in:

- A. Heap
- B. Stack
- C. Queue
- D. ROM

15. When does infinite recursion occur?

- A. Too many loops
- B. Base case missing
- C. No arguments
- D. Static variables used

16. What is the output?

```
void fun(int n){ if(n==0) return; fun(n-1); printf("%d", n); }  
fun(3);
```

- A. 123
- B. 321
- C. 111
- D. None

17. Which recursion is used for binary search?

- A. Linear
- B. Tail
- C. Divide-and-conquer
- D. Inline

18. Which recursive function is most expensive?

- A. Factorial
- B. Fibonacci
- C. GCD
- D. Sum of digits

19. A recursive function calling itself 3 times is:

- A. multi-recursion
- B. bi-recursion

- C. ternary recursion
  - D. cyclic recursion
20. Recursion with strings requires:
- A. char pointer
  - B. index
  - C. base condition
  - D. All

**Pointers:**

**1. What does a pointer store?**

- A. Address of a variable
- B. Value of a variable
- C. Both address and value
- D. Memory size of variable

**2. Which operator is used to access the value stored at an address?**

- A. &
- B. \*
- C. ->
- D. sizeof

**3. What is the output?**

```
int a = 10;  
int *p = &a;  
printf("%d", *p);
```

- A. 0
- B. 10
- C. Address of a
- D. Garbage

**4. Which one is a valid pointer declaration?**

- A. int p;
- B. int \*p;

- C. pointer p;
- D. address p;

**5. A pointer initialized to NULL means:**

- A. It stores 0 as a value
- B. It points to address 0
- C. It points to no valid memory location
- D. It stores garbage

**6. What is the size of a pointer on a 64-bit system?**

- A. 2 bytes
- B. 4 bytes
- C. 8 bytes
- D. Depends on data type

**7. What will this print?**

```
int arr[] = {2, 4, 6, 8};  
int *p = arr;  
printf("%d", *(p + 2));
```

- A. 2
- B. 4
- C. 6
- D. 8

**8. Which pointer arithmetic is allowed?**

- A. pointer + pointer
- B. pointer - pointer
- C. pointer + integer
- D. pointer \* integer

**9. Wild pointer means:**

- A. Pointer with NULL value
- B. Pointer not initialized



- C. Pointer pointing to freed memory
- D. Pointer to pointer

**10. What is the output?**

```
char *s = "Hello";  
printf("%c", *(s+1));
```

- A. H
- B. e
- C. l
- D. Undefined

**11. Which is correct for accessing array elements using pointers?**

- A. `arr[i] == *(arr + i)`
- B. `arr[i] != *(arr + i)`
- C. `arr[i] == (arr * i)`
- D. None

**12. What is the output?**

```
int x = 5;  
int *p = &x;  
(*p)++;  
printf("%d", x);
```

- A. 4
- B. 5
- C. 6
- D. Garbage

**13. Dangling pointer occurs when:**

- A. Pointer is NULL
- B. Pointer is uninitialized
- C. Memory is freed but pointer still points to it
- D. Pointer points to invalid type

**14. What is printed?**

```
int a = 10, b = 20;  
int *p = &a;  
p = &b;  
printf("%d", *p);
```

- A. 10
- B. 20
- C. Address of a
- D. Garbage

**15. Pointer to pointer stores:**

- A. Value of variable
- B. Address of variable
- C. Address of another pointer
- D. None

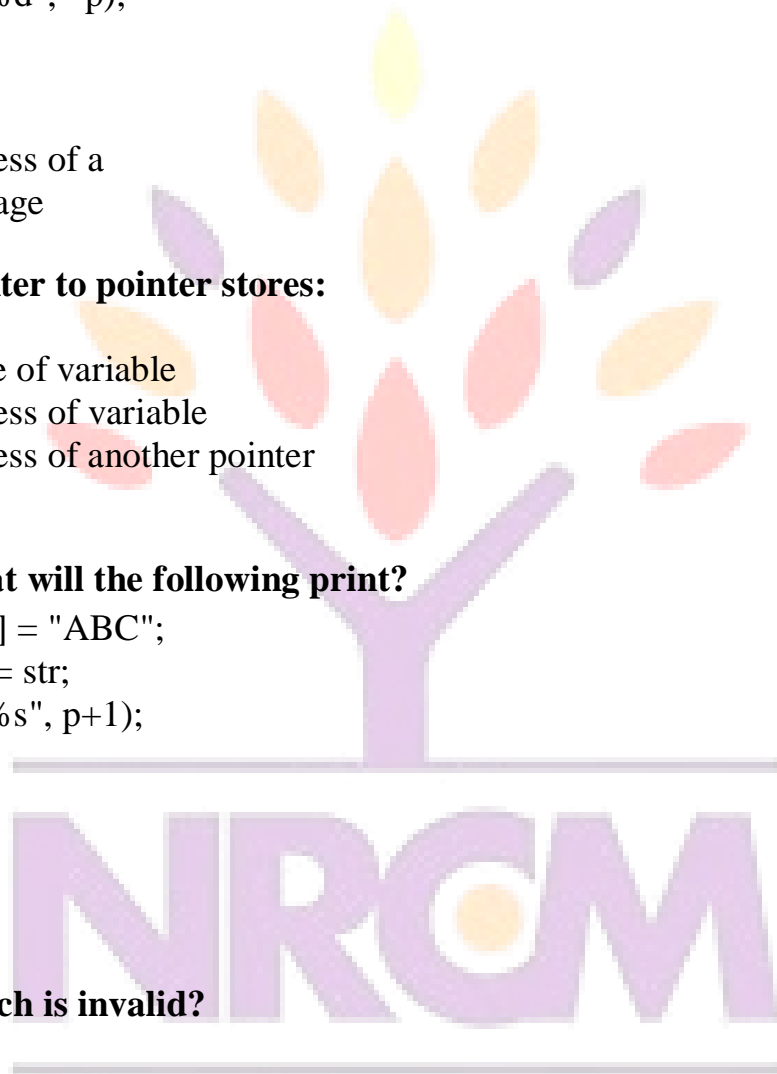
**16. What will the following print?**

```
char str[] = "ABC";  
char *p = str;  
printf("%s", p+1);
```

- A. ABC
- B. BC
- C. C
- D. Error

**17. Which is invalid?**

- A. `int *p;`
- B. `float q;`
- C. *double r;*
- D. `char *c;`



**18. Pointer subtraction returns:**

- A. Number of elements between two pointers
- B. Byte difference
- C. Address difference
- D. None

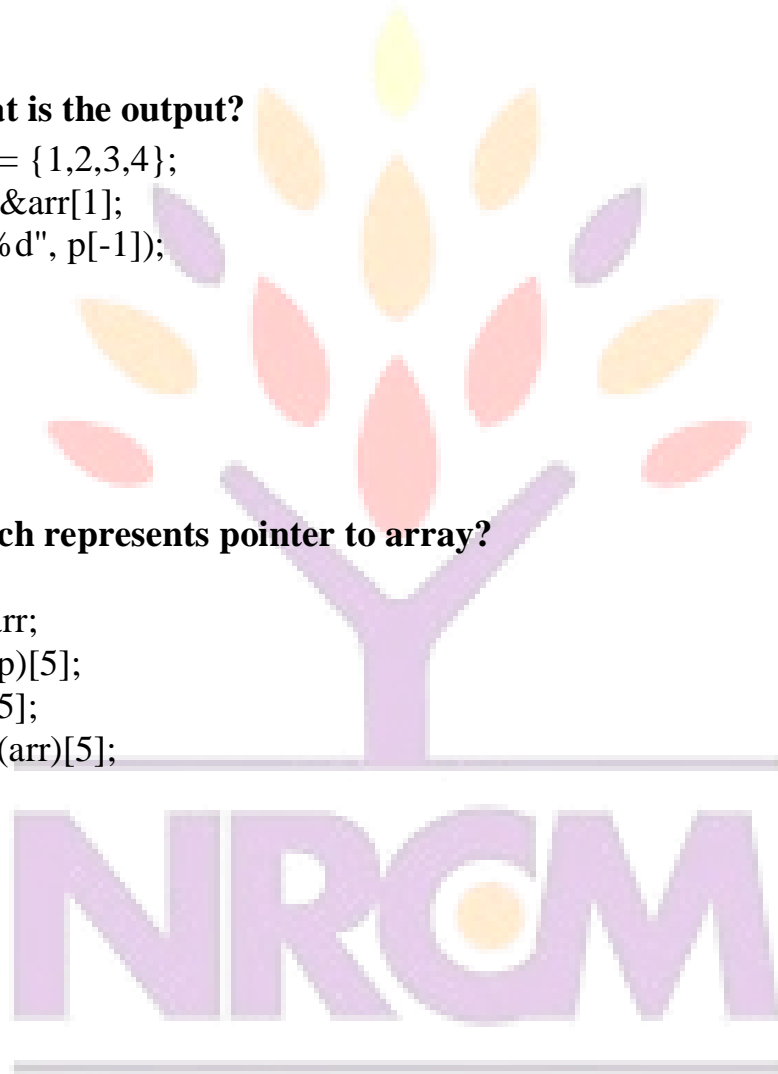
**19. What is the output?**

```
int arr[] = {1,2,3,4};  
int *p = &arr[1];  
printf("%d", p[-1]);
```

- A. 1
- B. 2
- C. 3
- D. Error

**20. Which represents pointer to array?**

- A. int \*arr;
- B. int (\*p)[5];
- C. int p[5];
- D. int &(arr)[5];



your roots to success...