

Fullstack Development Milestone Test-5

Round 1

1. If the elements in two arrays are related by their subscripts, the arrays are called as _____ arrays.

- a) associated
- b) coupled
- c) matching
- d) parallel

Correct Ans: parallel

2. The strStates and strCapitals arrays are parallel arrays. If Illinois is stored in the second element in the strStates array, where is its capital (Springfield) stored?

- a) strCapitals(1)
- b) strCapitals(2)
- c) strCapitals(3)
- d) strCapitals(4)

Correct Ans: strCapitals(1)

3. A _____ resembles a table.

- a)One-dimensional array
- b)Two-dimensional array
- c)Three-dimensional array
- d)N-dimensional array

Correct Ans: Two-dimensional array

4. We can determine number of elements in two-dimensional array by

- a)Multiplying number of rows and number of columns
- b)Adding number of rows and number of columns
- c)Multiplying number of rows and number of rows
- d)Adding number of columns and number of columns

Correct Ans: Multiplying number of rows and number of columns

5. The _____ in a two-dimensional array specifies the elements row and column position.

- a)Superscript
- b)Subscript
- c)Row number
- d)Column number

Correct Ans: Subscript

6. Each element in a two-dimensional array is identified by a unique combination of _____

- a) One subscript
- b) Two subscripts
- c) Three subscripts
- d) Zero subscript

Correct Ans: Two subscripts

7. The subscripts are _____ than the row and column in which the element is located.

- a) One number less
- b) One number more
- c) Two number less
- d) Two number more

Correct Ans: One number less

8. To traverse two dimensional array you require _____ loops.

- a) 1
- b) 2
- c) 3
- d) 4

Correct Ans: 2

9. Which of the following declares a two-dimensional array that has three rows and four columns?

- a)[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
- b)[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
- c)[[1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]]
- d)[[1, 2, 3], [1, 2, 3], [1, 2, 3]]

Correct Ans: [[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]

10. How to declare a two dimensional array?

- a)let a = [[1]]
- b)let a = []
- c)let a = [][]
- d)let a = [][][]

Correct Ans: let a = [[1]]

11. Recursion is a method in which the solution of a problem depends on

- a)Larger instances of different problems
- b)Larger instances of the same problem
- c)Smaller instances of the same problem
- d)Smaller instances of different problems

Correct Ans: Smaller instances of the same problem

12. Which of the following problems can't be solved using recursion?

- a) Factorial of a number
- b) Nth fibonacci number
- c) Length of a string
- d) Problems without base case

Correct Ans: Problems without base case

13. Recursion is similar to which of the following?

- a) Switch Case
- b) Loop
- c) If-else
- d) if elif else

Correct Ans: Loop

14. In recursion, the condition for which the function will stop calling itself is

-
- a) Best case
 - b) Worst case
 - c) Base case
 - d) There is no such condition

Correct Ans: Base case

15. What will happen when the below code snippet is executed?

```
function my_recursive_function()
{
my_recursive_function();
}
my_recursive_function();
```

- a)The code will be executed successfully and no output will be generated
- b)The code will be executed successfully and random output will be generated
- c)The code will show a compile time error
- d)The code will run for some time and stop when the stack overflows

Correct Ans: The code will run for some time and stop when the stack overflows

16. What is the output of the following code?

```
function my_recursive_function(n)
{
if(n == 0)
return;
console.log(n);
my_recursive_function(n-1);
}
function main()
{
```

```
my_recursive_function(10);  
return 0;  
}
```

a)10

b)1

c)10 9 8 ... 1 0

d)10 9 8 ... 1

Correct Ans: 10 9 8 ... 1

17. What is the base case for the following code?

```
function my_recursive_function(n)  
{  
  if(n == 0)  
    return;  
  console.log(n);  
  my_recursive_function(n-1);  
}  
  
function main()  
{  
  my_recursive_function(10);  
  return 0;  
}
```

- a) return
- b) console.log(n)
- c) if(n == 0)
- d) my_recursive_function(n-1)

Correct Ans: if(n == 0)

18. How many times is the recursive function called, when the following code is executed?

```
function my_recursive_function(n)
```

```
{
```

```
  if(n == 0)
```

```
  return;
```

```
  console.log(n);
```

```
  my_recursive_function(n-1);
```

```
}
```

```
function main()
```

```
{
```

```
  my_recursive_function(10);
```

```
  return 0;
```

```
}
```

a) 9

b) 10

c)11

d)12

Correct Ans: 11

19. What does the following recursive code do?

```
function my_recursive_function(n)
```

```
{
```

```
  if(n == 0)
```

```
    return;
```

```
    my_recursive_function(n-1);
```

```
    console.log(n);
```

```
}
```

```
function main()
```

```
{
```

```
  my_recursive_function(10);
```

```
  return 0;
```

```
}
```

a)Prints the numbers from 10 to 1

b)Prints the numbers from 10 to 0

c)Prints the numbers from 1 to 10

d)Prints the numbers from 0 to 10

Correct Ans: Prints the numbers from 1 to 10

20. Which of the following statements is true?

- a) Recursion is always better than iteration
- b) Recursion uses more memory compared to iteration
- c) Recursion uses less memory compared to iteration
- d) Iteration is always better and simpler than recursion

Correct Ans: Recursion uses more memory compared to iteration

21. What will be the output of the following code?

```
let cnt=0;

function my_recursive_function(n)
{
  if(n == 0)
    return;
  cnt++;
  my_recursive_function(n/10);
}

function main()
{
  my_recursive_function(123456789);
  console.log(cnt);
  return 0;
}
```

```
}
```

a)123456789

b)10

c)0

d)9

Correct Ans: 9

22. What will be the output of the following code?

```
function my_recursive_function(int n)
```

```
{
```

```
if(n == 0)
```

```
{
```

```
console.log("False");
```

```
return;
```

```
}
```

```
if(n == 1)
```

```
{
```

```
console.log("True");
```

```
return;
```

```
}
```

```
if(n%2==0)
```

```
my_recursive_function(n/2);
```

```
else  
{  
  console.log("False");  
  return;  
}
```

a)FALSE

b)TRUE

Correct Ans: FALSE

23. What is the output of the following code?

```
let cnt = 0;  
  
function my_recursive_function(s, i)  
{  
  if(s[i] == '\0')  
    return;  
  
  if(s[i] == 'a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'o' || s[i] == 'u')  
    cnt++;  
  
  my_recursive_function(s,i+1);  
}  
  
function main()  
{
```

my

a)6

b)9

c)5

d)10

Correct Ans: 6

24. What is the output of the following code?

```
function my_recursive_function(arr, val, idx, len)
```

```
{
```

```
  if(idx == len)
```

```
  {
```

```
    console.log("-1");
```

```
    return ;
```

```
  }
```

```
  if(arr[idx] == val)
```

```
  {
```

```
    console.log(idx);
```

```
    return;
```

```
  }
```

```
  my_recursive_function(arr,val,idx+1,len);
```

```
}
```

function main()

{

a)3

b)4

c)5

d)6

Correct Ans: 4

25. In general, which of the following methods isn't used to find the factorial of a number?

a)Recursion

b)Iteration

c)Dynamic programming

d)Non iterative / recursive

Correct Ans: Non iterative / recursive

26. Which of the following recursive formula can be used to find the factorial of a number?

a) $\text{fact}(n) = n * \text{fact}(n)$

b) $\text{fact}(n) = n * \text{fact}(n+1)$

c) $\text{fact}(n) = n * \text{fact}(n-1)$

d) $\text{fact}(n) = n * \text{fact}(1)$

Correct Ans: $\text{fact}(n) = n * \text{fact}(n-1)$

27. Consider the following iterative implementation to find the factorial of a number. Which of the lines should be inserted to complete the below code?

```
function main()  
{  
  let n = 6, i;  
  let fact = 1;  
  for(i=1;i<=n;i++)  
    _____;  
  console.log(fact);  
  return 0;  
}
```

- a) $\text{fact} = \text{fact} + i$
- b) $\text{fact} = \text{fact} * i$
- c) $i = i * \text{fact}$
- d) $i = i + \text{fact}$

Correct Ans: $\text{fact} = \text{fact} * i$

28. Consider the following recursive implementation to find the factorial of a number. Which of the lines should be inserted to complete the below code?

```
function fact(n)  
{
```

```
if(_____)
return 1;
return n * fact(n - 1);
}
```

```
function main()
```

```
{
```

```
let n = 5;
```

```
let ans
```

```
a)n = 0
```

```
b)n != 0
```

```
c)n == 0
```

```
d)n == 1
```

Correct Ans: n == 0

29. The time complexity of the following recursive implementation to find the factorial of a number is _____

```
function fact(n)
```

```
{
```

```
if(_____)
return 1;
```

```
return n * fact(n - 1);
```

```
}
```



```
function main()
{
let n = 5;
let ans = fact(n);
console.log(ans);
return 0;
}
```

a) $O(1)$

b) $O(n)$

c) $O(n^2)$

d) $O(n^3)$

Correct Ans: $O(n)$

30. What is the space complexity of the following recursive implementation to find the factorial of a number?

```
function fact(n)
{
if(_____)
return 1;
return n * fact(n - 1);
}

function main()
```

```
{  
let n = 5;  
let ans = fact(n);  
console.log(ans);  
return 0;  
}
```

a) $O(1)$

b) $O(n)$

c) $O(n^2)$

d) $O(n^3)$

Correct Ans: $O(1)$

31. Popular Notations in Complexity Analysis of Algorithms

a)Big-O Notation

b)Omega Notation

c)Theta Notation

d)All the above

Correct Ans: All the above

32. Which Complexity analysis is generally used?

a)Worst Case Analysis

b)Average Case Analysis

- c)Best Case Analysis
- d)All the above

Correct Ans: Worst Case Analysis

33. Properties of Asymptotic Notations?

- a)General Properties
- b)Basic Properties
- c)Non linear
- d)All the above

Correct Ans: General Properties

34. The data structures whose operations are analyzed using Amortized Analysis

- a)Hash Tables
- b)Disjoint Sets
- c)Splay Trees
- d)All the above

Correct Ans: All the above

35. How to solve Recurrence Relation for Complexity Analysis of Algorithms

- a)Substitution Method
- b)Non Substitution Method
- c)Greedy

d)All the above

Correct Ans: Substitution Method

36. Constant Time Complexity is

a) $O(1)$

b) $O(n)$

c) $O(nc)$

d) $O(\text{Log Log } n)$

Correct Ans: $O(1)$

37. Linear Time Complexity

a) $O(1)$

b) $O(n)$

c) $O(nc)$

d) $O(\text{Log Log } n)$

Correct Ans: $O(n)$

38. Quadratic Time Complexity

a) $O(1)$

b) $O(n)$

c) $O(n^c)$

d) $O(\text{Log Log } n)$

Correct Ans: $O(n^c)$

39. Logarithmic Time Complexity

- a) $O(1)$
- b) $O(n)$
- c) $O(n^c)$
- d) $O(\log \log n)$

Correct Ans: $O(\log \log n)$

40. What is Algorithm complexity and how to find it?

- a) Time Factor
- b) Space Factor
- c) None of the above
- d) All the above

Correct Ans: All the above