

## Introduction

**1. What is a data structure?**

- a) A way to store and organize data efficiently
- b) A type of database
- c) A programming language
- d) A method for writing algorithms

✓ **Answer:** a) A way to store and organize data efficiently

**2. Which of the following is NOT a linear data structure?**

- a) Stack
- b) Queue
- c) Tree
- d) Array

✓ **Answer:** c) Tree

**3. Which of the following operations has  $O(1)$  time complexity in an array?**

- a) Insertion at the end
- b) Searching an element
- c) Insertion at the beginning
- d) Deletion at the middle

✓ **Answer:** a) Insertion at the end

**4. Which data structure follows the Last In First Out (LIFO) principle?**

- a) Queue
- b) Stack
- c) Linked List
- d) Hash Table

✓ **Answer:** b) Stack

**5. What is an algorithm?**

- a) A programming language
- b) A flowchart
- c) A set of well-defined steps to solve a problem
- d) A computer program

✓ **Answer:** c) A set of well-defined steps to solve a problem

**6. What is the time complexity of searching an element in an unsorted array?**

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

d)  $O(n \log n)$

✓ Answer: c)  $O(n)$

**7. Which of the following notations is the worst-case time complexity?**

a) Big-O ( $O$ )

b) Big-Omega ( $\Omega$ )

c) Big-Theta ( $\Theta$ )

d) Small-O ( $o$ )

✓ Answer: a) Big-O ( $O$ )

**8. What is the best case time complexity of QuickSort?**

a)  $O(n)$

b)  $O(n \log n)$

c)  $O(n^2)$

d)  $O(\log n)$

✓ Answer: b)  $O(n \log n)$

**9. Which notation represents the average-case complexity?**

a)  $O(n)$

b)  $\Theta(n)$

c)  $\Omega(n)$

d)  $o(n)$

✓ Answer: b)  $\Theta(n)$

**10. If an algorithm has a complexity of  $O(\log n)$ , how does its execution time change when input size doubles?**

a) Doubles

b) Increases logarithmically

c) Remains constant

d) Becomes linear

✓ Answer: b) Increases logarithmically

**11. Which sorting algorithm has the worst-case time complexity of  $O(n^2)$ ?**

a) Merge Sort

b) Quick Sort

c) Bubble Sort

d) Radix Sort

✓ Answer: c) Bubble Sort

**12. What is the time complexity of Binary Search?**

a)  $O(n)$

b)  $O(\log n)$

c)  $O(n \log n)$

d)  $O(n^2)$

✓ Answer: b)  $O(\log n)$

**13. Which of the following sorting algorithms is NOT based on comparison?**

a) Merge Sort

b) Quick Sort

c) Bubble Sort

d) Counting Sort

✓ Answer: d) Counting Sort

**14. Which sorting algorithm is best for nearly sorted data?**

a) Quick Sort

b) Bubble Sort

c) Insertion Sort

d) Merge Sort

✓ Answer: c) Insertion Sort

**15. Which search algorithm works efficiently on sorted data?**

a) Linear Search

b) Binary Search

c) Breadth-First Search

d) Depth-First Search

✓ Answer: b) Binary Search

### **ARRAY:-**

**1. What is the default value of an array of int in Java?**

a) 1

b) 0

c) Null

d) Garbage value

✓ Answer: b) 0

**2. How do you declare an array in Java?**

a) `int arr = new int[10];`

b) `int arr[] = new int[10];`

c) `int[] arr = new int[10];`

d) Both b and c

✓ Answer: d) Both b and c

**3. What will happen if an array index goes out of bounds?**

a) The program will terminate abruptly.

b) The compiler will throw an error.

c) It will return a garbage value.

d) It will continue execution normally.

✓ **Answer:** a) The program will terminate abruptly.

**4. What is the index of the first element of an array in Java?**

a) 0

b) 1

c) -1

d) None of the above

✓ **Answer:** a) 0

**5. What will be the output of the following code?**

```
int arr[] = {1, 2, 3, 4, 5};  
System.out.println(arr.length);
```

a) 4

b) 5

c) 6

d) Compiler error

✓ **Answer:** b) 5

**6. How do you initialize a two-dimensional array in Java?**

a) `int[][] arr = new int[3][3];`

b) `int arr[][] = new int[3,3];`

c) `int arr[3][3] = new int[][];`

d) `int arr = new int[3][3];`

✓ **Answer:** a) `int[][] arr = new int[3][3];`

**7. What will be the output of the following code?**

```
int arr[] = new int[5];  
System.out.println(arr[2]);
```

a) 0

b) 2

c) Compilation error

d) Runtime error

✓ **Answer:** a) 0

**8. What is the correct way to iterate over an array using an enhanced for loop?**

a) `for(int i=0; i<arr.length; i++)`

b) `for(int element : arr)`

c) `for(int i : arr.length)`

d) `while(arr.length > 0)`

✓ Answer: b) `for(int element : arr)`

**9. What is the time complexity of accessing an element in an array?**

a)  $O(1)$

b)  $O(n)$

c)  $O(\log n)$

d)  $O(n^2)$

✓ Answer: a)  $O(1)$

**10. How can you copy an array in Java?**

a) `System.arraycopy()`

b) `Arrays.copyOf()`

c) `clone()`

d) All of the above

✓ Answer: d) All of the above

**11. Can an array store different data types in Java?**

a) Yes

b) No

✓ Answer: b) No

**12. What is returned by `Arrays.binarySearch()` when the element is not found?**

a) -1

b) -2

c) A negative index

d) 0

✓ Answer: c) A negative index

**13. What will be the output of the following code?**

```
int arr[] = {1, 2, 3, 4};  
System.out.println(arr[arr.length - 1]);
```

a) 1

b) 4

c) Compilation error

d) Runtime error

✓ Answer: b) 4

**14. Which method is used to sort an array in Java?**

a) `Arrays.sort()`

b) `Collections.sort()`

c) `Sort.sort()`

d) array.sort()

✓ Answer: a) Arrays.sort()

**15. How can you convert an array to a list in Java?**

a) Arrays.toList()

b) Arrays.asList()

c) Array.asList()

d) List.toArray()

✓ Answer: b) Arrays.asList()

**16. What is the default value of a boolean array in Java?**

a) true

b) false

c) null

d) Compiler error

✓ Answer: b) false

**17. What will be the output of this code?**

```
String arr[] = new String[5];
```

```
System.out.println(arr[0]);
```

a) Empty string ""

b) null

c) Compiler error

d) Garbage value

✓ Answer: b) null

**18. What does Arrays.fill(arr, 5) do?**

a) Fills the array arr with the value 5

b) Fills only the first five elements of the array

c) Throws an exception

d) Does nothing

✓ Answer: a) Fills the array arr with the value 5

**19. Can you increase the size of an array after it is declared?**

a) Yes

b) No

✓ Answer: b) No

**20. How can you dynamically resize an array in Java?**

a) Using resize() method

b) By reassigning it to a larger array

c) Using increaseArraySize()

d) Java arrays cannot be resized

✓ **Answer:** b) By reassigning it to a larger array

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

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {10, 20, 30, 40};  
        System.out.println(arr[2]);  
    }  
}
```

A) 10

B) 20

C) 30

D) 40

**Answer:** C) 30

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

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = new int[5];  
        System.out.println(arr[3]);  
    }  
}
```

A) Compilation error

B) Runtime error

C) 0

D) Garbage value

**Answer:** C) 0

**23. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3, 4, 5};  
        System.out.println(arr.length);  
    }  
}
```

- A) 4
- B) 5
- C) Compilation error
- D) Runtime error

**Answer:** B) 5

**24. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {5, 10, 15};  
        arr[1] = 20;  
        System.out.println(arr[1]);  
    }  
}
```

- A) 10
- B) 20
- C) Compilation error
- D) Runtime error

**Answer:** B) 20

**25. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3, 4};  
        System.out.println(arr[arr.length - 1]);  
    }  
}
```

- A) 1
- B) 2
- C) 3
- D) 4

**Answer:** D) 4

**26. What is the default value of a boolean array element in Java?**

- A) 0
- B) false



- C) true
- D) null

**Answer:** B) false

**27. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[][] arr = {{1, 2}, {3, 4}};  
        System.out.println(arr[1][1]);  
    }  
}
```

- A) 1
- B) 2
- C) 3
- D) 4

**Answer:** D) 4

**28. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[][] arr = new int[2][2];  
        System.out.println(arr[0][1]);  
    }  
}
```

- A) Compilation error
- B) Runtime error
- C) 0
- D) Garbage value

**Answer:** C) 0

**29. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {10, 20, 30, 40};  
        System.out.println(arr[-1]);  
    }  
}
```

- A) 10
- B) 40
- C) Compilation error
- D) Runtime error

**Answer:** D) Runtime error

**30. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {10, 20, 30, 40};  
        System.out.println(arr[arr.length]);  
    }  
}
```

- A) Compilation error
- B) 40
- C) Runtime error
- D) 30

**Answer:** C) Runtime error

**31. How do you declare an array of 5 integers in Java?**

- A) `int[5] arr;`
- B) `int arr[5];`
- C) `int[] arr = new int[5];`
- D) `int arr = new int[5];`

**Answer:** C) `int[] arr = new int[5];`

**32. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String[] names = {"Alice", "Bob", "Charlie"};  
        System.out.println(names[1]);  
    }  
}
```

- A) Alice
- B) Bob
- C) Charlie
- D) Compilation error

**Answer:** B) Bob

**33. Which of the following is true about Java arrays?**

- A) Arrays can store only primitive data types
- B) Arrays are dynamically resizable
- C) Arrays have a fixed size
- D) Arrays do not allow duplicate values

**Answer:** C) Arrays have a fixed size

**34. What is the default value of an uninitialized integer array element in Java?**

- A) 0
- B) null
- C) Garbage value
- D) Compilation error

**Answer:** A) 0

**35. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = new int[3];  
        arr[0] = 10;  
        arr[1] = 20;  
        arr[2] = 30;  
        for(int i = 0; i < arr.length; i++)  
            System.out.print(arr[i] + " ");  
    }  
}
```

- A) 10 20 30
- B) 10 20
- C) 30 20 10
- D) Compilation error

**Answer:** A) 10 20 30

**36. What will happen if an array element is accessed beyond its size?**

- A) Compilation error
- B) Runtime error (ArrayIndexOutOfBoundsException)

- C) The value will be 0
- D) It will return null

**Answer:** B) Runtime error (ArrayIndexOutOfBoundsException)

**37. How do you initialize an array in Java?**

- A) `int[] arr = new int[3];`
- B) `int[] arr = {1, 2, 3};`
- C) `int arr[] = new int[]{1, 2, 3};`
- D) All of the above

**Answer:** D) All of the above

**38. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {5, 10, 15, 20};  
        System.out.println(arr.length);  
    }  
}
```

- A) 3
- B) 4
- C) 5
- D) Compilation error

**Answer:** B) 4

**39. What is the time complexity of accessing an element in an array by index?**

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** A)  $O(1)$

**40. What is the output of this program?**

```
public class Main {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3};  
        System.out.println(arr instanceof Object);  
    }  
}
```

```
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** A) true

## String

**1. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Hello";  
        System.out.println(s.length);  
    }  
}
```

- A) 4
- B) 5
- C) 6
- D) Compilation error

**Answer:** D

**2. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.charAt(2));  
    }  
}
```

- A) J
- B) a
- C) v
- D) Compilation error

**Answer:** C) v

**3. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s1 = "Hello";  
        String s2 = "Hello";  
        System.out.println(s1 == s2);  
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** A) true

**4. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s1 = new String("Hello");  
        String s2 = new String("Hello");  
        System.out.println(s1 == s2);  
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** B) false

**5. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Hello";  
        s = s.concat(" World");  
        System.out.println(s);  
    }  
}
```

- A) Hello
- B) Hello World

- C) World
- D) Compilation error

**Answer:** B) Hello World

**6. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Hello";  
        System.out.println(s.toUpperCase());  
    }  
}
```

- A) hello
- B) HELLO
- C) HeLlO
- D) Compilation error

**Answer:** B) HELLO

**7. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = " Java ";  
        System.out.println(s.trim());  
    }  
}
```

- A) " Java "
- B) "Java"
- C) " Java"
- D) Compilation error

**Answer:** B) "Java"

**8. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.substring(1, 3));  
    }  
}
```

- A) Ja
- B) av
- C) va
- D) Compilation error

**Answer:** B) av

**9. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.replace('a', 'o'));  
    }  
}
```

- A) Jovo
- B) Jova
- C) Jovoa
- D) Compilation error

**Answer:** A) Jovo

**10. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.contains("va"));  
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** A) true

**11. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.indexOf('a'));  
    }  
}
```



}

- A) 1
- B) 2
- C) 3
- D) -1

**Answer:** A) 1

**12. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.startsWith("J"));  
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** A) true

**13. Which of the following creates an empty string in Java?**

- A) String s = "";
- B) String s = new String();
- C) Both A and B
- D) None of the above

**Answer:** C) Both A and B

**14. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s1 = "Hello";  
        String s2 = "hello";  
        System.out.println(s1.equalsIgnoreCase(s2));  
    }  
}
```

- A) true
- B) false

- C) Compilation error
- D) Runtime error

**Answer:** A) true

**15. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java";  
        System.out.println(s.isEmpty());  
    }  
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** B) false

**16. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Java Programming";  
        System.out.println(s.split(" ")[0]);  
    }  
}
```

- A) Java
- B) Programming
- C) Java Programming
- D) Compilation error

**Answer:** A) Java

**17. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "Hello";  
        System.out.println(s.concat(null));  
    }  
}
```

- A) Hello
- B) nullHello
- C) Compilation error
- D) Runtime error

**Answer:** D) Runtime error

**18. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder("Java");  
        sb.append(" Programming");  
        System.out.println(sb);  
    }  
}
```

- A) Java
- B) Java Programming
- C) Compilation error
- D) Runtime error

**Answer:** B) Java Programming

**19. Which of the following is true about Java Strings?**

- A) Strings are mutable
- B) Strings are immutable
- C) Strings cannot be created using new
- D) Strings are stored in stack memory

**Answer:** B) Strings are immutable

**20. What will be the output of the following code?**

```
public class Main {  
    public static void main(String[] args) {  
        String s = "hello";  
        s = s.replace('l', 'w');  
        System.out.println(s);  
    }  
}
```

- A) hello
- B) hewwo
- C) hwwwwo
- D) Compilation error

**Answer:** B) hewwo

**21. Which of the following is used to declare a String in Java?**

- a) `String str = "Hello";`
- b) `String str = new String("Hello");`
- c) Both a and b
- d) None of the above

✓ **Answer:** c) Both a and b

**22. What is the output of the following code?**

```
String str = "Java";  
System.out.println(str.length());
```

- a) 3
- b) 4
- c) 5
- d) Compiler error

✓ **Answer:** b) 4

**23. Strings in Java are stored in?**

- a) Heap memory
- b) Stack memory
- c) String pool (inside Heap memory)
- d) Register memory

✓ **Answer:** c) String pool (inside Heap memory)

**24. What will be the output of the following code?**

```
String s1 = "Java";  
String s2 = "Java";  
System.out.println(s1 == s2);
```

- a) true
- b) false
- c) Compilation error
- d) Runtime error

✓ **Answer:** a) true

**25. What is the correct way to concatenate two strings in Java?**

- a) `str1.concat(str2);`
- b) `str1 + str2;`
- c) `str1.append(str2);`
- d) Both a and b

✓ **Answer:** d) Both a and b

**26. What is the output of `str.toUpperCase()` if `str = "hello";`?**

- a) HELLO
- b) hello
- c) Hello
- d) Compiler error

✓ **Answer:** a) HELLO

**27. What does `str.trim()` do?**

- a) Removes spaces from the start
- b) Removes spaces from the end
- c) Removes spaces from both ends
- d) Does nothing

✓ **Answer:** c) Removes spaces from both ends

**28. What is the return type of `charAt(int index)`?**

- a) char
- b) int
- c) String
- d) boolean

✓ **Answer:** a) char

**29. What is the output of `System.out.println("hello".substring(1,3));`?**

- a) he
- b) el
- c) ello
- d) Compilation error

✓ **Answer:** b) el

**30. How do you compare two strings for equality in Java?**

- a) `str1 == str2`
- b) `str1.equals(str2)`
- c) `str1.compareTo(str2) == 0`
- d) Both b and c

✓ **Answer:** d) Both b and c

**31. What will be the output of `System.out.println("Java".indexOf('a'));`?**

- a) 1

- b) 2
- c) -1
- d) Compilation error

✓ Answer: a) 1

**32. What does the replace() method do in Java Strings?**

- a) Replaces all occurrences of a character with another
- b) Modifies the original string
- c) Replaces only the first occurrence of a character
- d) Throws an exception

✓ Answer: a) Replaces all occurrences of a character with another

**33. Which method is used to split a string into an array?**

- a) split()
- b) substring()
- c) cut()
- d) divide()

✓ Answer: a) split()

**34. What is the return type of split() method?**

- a) String
- b) String[]
- c) List<String>
- d) int

✓ Answer: b) String[]

**35. Which method converts a string to a character array?**

- a) toCharArray()
- b) toArray()
- c) split()
- d) charArray()

✓ Answer: a) toCharArray()

**36. Why are strings immutable in Java?**

- a) Security reasons
- b) Performance optimization
- c) String pooling
- d) All of the above

✓ Answer: d) All of the above

**37. What is the main advantage of StringBuilder over String?**

- a) Faster performance for modifications
- b) Thread safety
- c) Uses less memory

d) None of the above

✓ **Answer:** a) Faster performance for modifications

**38. Which class provides a mutable version of String?**

a) StringBuffer

b) StringBuilder

c) Both a and b

d) None of the above

✓ **Answer:** c) Both a and b

**39. What is the difference between StringBuilder and StringBuffer?**

a) StringBuffer is synchronized, StringBuilder is not

b) StringBuilder is synchronized, StringBuffer is not

c) Both are the same

d) None of the above

✓ **Answer:** a) StringBuffer is synchronized, StringBuilder is not

**40. What is the correct way to append text in StringBuilder?**

a) sb.append("text");

b) sb.add("text");

c) sb.concatenate("text");

d) sb.plus("text");

✓ **Answer:** a) sb.append("text");

## Linked List.

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

```
class Node {
    int data;
    Node next;
    Node(int data) {
        this.data = data;
        this.next = null;
    }
}

public class Main {
    public static void main(String[] args) {
        Node head = new Node(10);
        head.next = new Node(20);
        head.next.next = new Node(30);
        System.out.println(head.next.data);
    }
}
```

```
}
```

- A) 10
- B) 20
- C) 30
- D) Compilation error

**Answer:** B) 20

**2. What will be the output of the following code?**

```
public class Main {  
    static class Node {  
        int data;  
        Node next;  
        Node(int data) {  
            this.data = data;  
            this.next = null;  
        }  
    }  
    public static void main(String[] args) {  
        Node head = new Node(10);  
        head.next = new Node(20);  
        head = head.next;  
        System.out.println(head.data);  
    }  
}
```

- A) 10
- B) 20
- C) null
- D) Compilation error

**Answer:** B) 20

**3. What will be the output of the following code?**

```
class Node {  
    int data;  
    Node next;  
    Node(int data) {  
        this.data = data;  
        this.next = null;  
    }  
}  
public class Main {
```



```

    public static void main(String[] args) {
        Node head = new Node(10);
        head.next = new Node(20);
        head.next.next = new Node(30);
        System.out.println(head.next.next.next);
    }
}

```

- A) 30
- B) null
- C) Compilation error
- D) Runtime error

**Answer:** B) null

4. What is the time complexity of inserting an element at the head of a singly linked list?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** A)  $O(1)$

5. What is the time complexity of inserting an element at the tail of a singly linked list (without a tail pointer)?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

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

```

public class Main {
    static class Node {
        int data;
        Node next;
        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }
}

```

```
public static void main(String[] args) {  
    Node head = new Node(5);  
    head.next = new Node(10);  
    head.next.next = new Node(15);  
    System.out.println(head.next.next.data);  
}  
}
```

- A) 5
- B) 10
- C) 15
- D) Compilation error

**Answer:** C) 15

7. What is the advantage of a doubly linked list over a singly linked list?

- A) Requires less memory
- B) Faster searching
- C) Can be traversed in both directions
- D) Uses less pointers

**Answer:** C) Can be traversed in both directions

8. What is the default value of a Node's next reference in Java?

- A) 0
- B) null
- C) false
- D) Garbage value

**Answer:** B) null

9. What is the time complexity of deleting a node from the middle of a singly linked list (given only its reference)?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

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

```
public class Main {  
    static class Node {  
        int data;  
        Node next;  
        Node(int data) {  
            this.data = data;  
            this.next = null;  
        }  
    }  
    public static void main(String[] args) {  
        Node head = new Node(1);  
        head.next = head;  
        System.out.println(head.next.data);  
    }  
}
```

- A) 1
- B) null
- C) Compilation error
- D) Infinite loop

**Answer:** A) 1

11. What is the time complexity of searching for an element in a singly linked list?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

12. What is the time complexity of inserting an element at the tail of a doubly linked list (with a tail pointer)?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** A)  $O(1)$

13. What will happen if we access an index greater than the size of the LinkedList?

- A) Returns null
- B) Throws an IndexOutOfBoundsException
- C) Prints 0
- D) Compiles but runs infinitely

**Answer:** B) Throws an IndexOutOfBoundsException

14. What will be the output of this code?

```
public class Main {  
    static class Node {  
        int data;  
        Node next;  
        Node(int data) {  
            this.data = data;  
            this.next = null;  
        }  
    }  
    public static void main(String[] args) {  
        Node head = new Node(5);  
        head.next = new Node(10);  
        System.out.println(head.next.next);  
    }  
}
```

- A) 5
- B) 10
- C) null
- D) Compilation error

**Answer:** C) null

15. What is the space complexity of a singly linked list with n nodes?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

16. Which of the following statements about circular linked lists is true?

- A) The last node points to null
- B) The last node points to the head
- C) It cannot have more than two nodes
- D) It always contains an even number of nodes

**Answer:** B) The last node points to the head

17. What is the time complexity of reversing a singly linked list?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

18. What will be the output of this code?

```
public class Main {
    static class Node {
        int data;
        Node next;
        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }
    public static void main(String[] args) {
        Node head = new Node(1);
        head.next = head;
        System.out.println(head.next == head);
    }
}
```

- A) true
- B) false
- C) Compilation error
- D) Runtime error

**Answer:** A) true

19. What is the main drawback of a singly linked list?

- A) It requires extra memory
- B) It cannot be traversed backward
- C) It has  $O(n)$  insertion at the head
- D) None of the above

**Answer:** B) It cannot be traversed backward

20. What is the time complexity of deleting the last node of a singly linked list (without a tail pointer)?

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

21. What is a Linked List in Java?

- a) A data structure that stores elements in contiguous memory locations
- b) A linear data structure where each element is a separate object
- c) A collection that cannot grow dynamically
- d) None of the above

✓ **Answer:** b) A linear data structure where each element is a separate object

22. What is the key advantage of a Linked List over an array?

- a) Random access is faster
- b) Memory allocation is contiguous
- c) Dynamic size and efficient insertions/deletions
- d) Linked Lists are always smaller in size

✓ **Answer:** c) Dynamic size and efficient insertions/deletions

23. Which of the following is not a type of Linked List?

- a) Singly Linked List
- b) Doubly Linked List
- c) Circular Linked List
- d) Indexed Linked List

✓ **Answer:** d) Indexed Linked List

**24. Which class in Java provides an implementation of a Linked List?**

- a) ArrayList
- b) LinkedList
- c) HashSet
- d) Vector

✓ **Answer:** b) LinkedList

**25. What is the time complexity of inserting an element at the beginning of a Linked List?**

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

✓ **Answer:** a)  $O(1)$

**26. How do you remove the last element of a LinkedList in Java?**

- a) `list.deleteLast();`
- b) `list.remove(list.size() - 1);`
- c) `list.removeLast();`
- d) `list.popLast();`

✓ **Answer:** c) `list.removeLast();`

**27. What does `list.getFirst()` do in a LinkedList?**

- a) Removes the first element
- b) Retrieves the first element without removing it
- c) Returns null
- d) Throws an error if the list is empty

✓ **Answer:** b) Retrieves the first element without removing it

**28. What is the worst-case time complexity for searching an element in a Linked List?**

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

✓ **Answer:** c)  $O(n)$

**29. How can you convert a LinkedList to an ArrayList in Java?**

- a) `ArrayList al = new ArrayList(linkedList);`
- b) `ArrayList al = linkedList.toArray();`
- c) `ArrayList al = linkedList.toList();`
- d) `ArrayList al = new ArrayList<>(linkedList);`

✓ **Answer:** d) `ArrayList al = new ArrayList<>(linkedList);`

**30. What is the default initial size of a LinkedList in Java?**

- a) 10
- b) 0
- c) 1
- d) Undefined

✓ **Answer:** b) 0

**31. What is the main difference between a Singly Linked List and a Doubly Linked List?**

- a) Singly Linked List stores only one element
- b) Doubly Linked List has references to both next and previous nodes
- c) Singly Linked List cannot store duplicate values
- d) None of the above

✓ **Answer:** b) Doubly Linked List has references to both next and previous nodes

**32. What is a Circular Linked List?**

- a) A Linked List that connects back to itself
- b) A Linked List with a fixed size
- c) A Linked List that cannot store duplicate values
- d) None of the above

✓ **Answer:** a) A Linked List that connects back to itself

**33. How do you check if a given Linked List is circular?**

- a) Using a HashSet to track visited nodes
- b) Using Floyd's cycle-finding algorithm
- c) Using a slow and fast pointer approach
- d) All of the above

✓ **Answer:** d) All of the above

**34. What is the time complexity of reversing a Doubly Linked List?**

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

✓ **Answer:** b)  $O(n)$

**35. What is the time complexity for inserting a node at the tail of a Circular Linked List?**

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

✓ **Answer:**



**36. What is a Linked List?**

- a) A linear data structure where elements are stored at contiguous memory locations
- b) A linear data structure where elements are connected using pointers
- c) A non-linear data structure
- d) A dynamic array

✓ **Answer:** b) A linear data structure where elements are connected using pointers

**37. What are the basic types of Linked Lists?**

- a) Singly Linked List
- b) Doubly Linked List
- c) Circular Linked List
- d) All of the above

✓ **Answer:** d) All of the above

**38. What is the main disadvantage of a Linked List over an array?**

- a) Dynamic size
- b) Requires more memory due to pointers
- c) Faster access to elements
- d) Easier to use

✓ **Answer:** b) Requires more memory due to pointers

**39. What does each node of a singly linked list contain?**

- a) Data only
- b) Pointer only
- c) Data and pointer to next node
- d) Data and pointers to both previous and next nodes

✓ **Answer:** c) Data and pointer to next node

**40. How do you insert an element at the beginning of a singly linked list?**

- a) Traverse to the last node and insert
- b) Update head to point to the new node and the new node's next to the old head
- c) Delete the head node first
- d) None of the above

✓ **Answer:** b) Update head to point to the new node and the new node's next to the old head

**41. What is the time complexity for inserting a node at the beginning of a Linked List?**

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

d)  $O(n \log n)$

✓ Answer: a)  $O(1)$

**42. What is the time complexity of inserting a node at the end of a Linked List (without tail pointer)?**

a)  $O(1)$

b)  $O(n)$

c)  $O(\log n)$

d)  $O(n \log n)$

✓ Answer: b)  $O(n)$

**43. What is the worst-case time complexity for searching an element in a singly linked list?**

a)  $O(1)$

b)  $O(\log n)$

c)  $O(n)$

d)  $O(n \log n)$

✓ Answer: c)  $O(n)$

**44. How can a linked list be efficiently reversed?**

a) Using a stack

b) Using recursion

c) Iterative method by changing the next pointers

d) All of the above

✓ Answer: d) All of the above

**45. How do you delete a node from a singly linked list?**

a) Remove the last node always

b) Modify the next pointer of the previous node to skip the node to be deleted

c) Replace it with NULL

d) None of the above

✓ Answer: b) Modify the next pointer of the previous node to skip the node to be deleted

**46. What is an advantage of a doubly linked list over a singly linked list?**

a) Faster search

b) Uses less memory

c) Can be traversed in both directions

d) Cannot be modified

✓ Answer: c) Can be traversed in both directions

**47. What additional pointer is present in a doubly linked list compared to a singly linked list?**

- a) Random pointer
  - b) Previous pointer
  - c) Tail pointer
  - d) Circular pointer
- ✓ Answer: b) Previous pointer

**48. What is the time complexity for inserting a node in the middle of a doubly linked list?**

- a)  $O(1)$
  - b)  $O(n)$
  - c)  $O(\log n)$
  - d)  $O(n \log n)$
- ✓ Answer: b)  $O(n)$

**49. What is the time complexity for deleting a node from a doubly linked list?**

- a)  $O(1)$  if node address is given
  - b)  $O(n)$  if node address is not given
  - c) Both a and b
  - d)  $O(n \log n)$
- ✓ Answer: c) Both a and b

**50. Which of the following operations requires modifying two pointers in a doubly linked list?**

- a) Insertion at the beginning
  - b) Insertion at the end
  - c) Deletion of a node
  - d) All of the above
- ✓ Answer: d) All of the above

**51. What is a circular linked list?**

- a) A list where the last node points back to the first node
  - b) A list where each node points to itself
  - c) A list that can only grow but not shrink
  - d) None of the above
- ✓ Answer: a) A list where the last node points back to the first node

**52. How do you traverse a circular linked list?**

- a) Using recursion
  - b) Using a do-while loop
  - c) Using a queue
  - d) Using an extra pointer
- ✓ Answer: b) Using a do-while loop

**53. What is the time complexity of inserting a node at the beginning of a circular linked list?**

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

✓ Answer: a)  $O(1)$

**54. What is a disadvantage of a circular linked list?**

- a) Cannot be implemented in Java
- b) Cannot traverse easily using a while loop
- c) Takes more memory than a singly linked list
- d) More complex insertion and deletion operations

✓ Answer: d) More complex insertion and deletion operations

**55. What is the time complexity of deleting the last node in a circular linked list?**

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

✓ Answer: b)  $O(n)$

### Recursion:-

**1. What will be the output of the following code?**

```
public class Main {  
    public static void fun(int n) {  
        if (n == 0) return;  
        System.out.print(n + " ");  
        fun(n - 1);  
    }  
    public static void main(String[] args) {  
        fun(5);  
    }  
}
```

- A) 1 2 3 4 5
- B) 5 4 3 2 1
- C) 5 4 3 2
- D) Infinite loop

**Answer:** B) 5 4 3 2 1

**2. What will be the output of the following recursive function?**

```
public class Main {  
    public static void fun(int n) {  
        if (n == 0) return;  
        fun(n - 1);  
        System.out.print(n + " ");  
    }  
    public static void main(String[] args) {  
        fun(3);  
    }  
}
```

- A) 3 2 1
- B) 1 2 3
- C) 3 2 1 0
- D) Compilation error

**Answer:** B) 1 2 3

**3. What is the base case in the following recursive function?**

```
public class Main {  
    public static int sum(int n) {  
        if (n == 1) return 1;  
        return n + sum(n - 1);  
    }  
    public static void main(String[] args) {  
        System.out.println(sum(5));  
    }  
}
```

- A) if (n == 0) return 0;
- B) if (n == 1) return 1;
- C) return n + sum(n - 1);
- D) No base case

**Answer:** B) if (n == 1) return 1;

**4. What will be the output of the following recursive function?**

```
public class Main {  
    public static int factorial(int n) {  
        if (n == 0) return 1;
```

```

        return n * factorial(n - 1);
    }
    public static void main(String[] args) {
        System.out.println(factorial(4));
    }
}

```

- A) 24
- B) 10
- C) 4
- D) Compilation error

**Answer:** A) 24

**5. What is the output of the following recursive function?**

```

public class Main {
    public static int fun(int n) {
        if (n <= 1) return n;
        return fun(n - 1) + fun(n - 2);
    }
    public static void main(String[] args) {
        System.out.println(fun(5));
    }
}

```

- A) 5
- B) 8
- C) 10
- D) 12

**Answer:** B) 8

**6. What is the time complexity of the above Fibonacci function?**

- A)  $O(n)$
- B)  $O(2^n)$
- C)  $O(n^2)$
- D)  $O(\log n)$

**Answer:** B)  $O(2^n)$

**7. What will be the output of the following recursive function?**

```

public class Main {
    public static void fun(int n) {

```

```

        if (n == 0) return;
        System.out.print(n + " ");
        fun(n / 2);
    }
    public static void main(String[] args) {
        fun(10);
    }
}

```

- A) 10 5 2 1
- B) 10 5 2 1 0
- C) 10 5 2
- D) 10 2 1

**Answer:** A) 10 5 2 1

**8. What happens when a recursive function lacks a base case?**

- A) Runs indefinitely
- B) Throws an exception
- C) Runs but produces no output
- D) Compiles but does nothing

**Answer:** A) Runs indefinitely

**9. What will be the output of the following recursive function?**

```

public class Main {
    public static void fun(int n) {
        if (n <= 0) return;
        fun(n - 1);
        System.out.print(n + " ");
        fun(n - 1);
    }
    public static void main(String[] args) {
        fun(3);
    }
}

```

- A) 3 2 1
- B) 1 2 1 3 1 2 1
- C) 1 2 3 2 1
- D) Compilation error

**Answer:** B) 1 2 1 3 1 2 1

**10. What will be the output of this tail-recursive function?**

```
public class Main {  
    public static void fun(int n) {  
        if (n == 0) return;  
        System.out.print(n + " ");  
        fun(n - 1);  
    }  
    public static void main(String[] args) {  
        fun(4);  
    }  
}
```

- A) 4 3 2 1
- B) 1 2 3 4
- C) 4 3 2 1 0
- D) Compilation error

**Answer:** A) 4 3 2 1

**11. What is the output of the following function?**

```
public class Main {  
    public static int fun(int n) {  
        if (n == 1) return 1;  
        return 2 * fun(n - 1);  
    }  
    public static void main(String[] args) {  
        System.out.println(fun(4));  
    }  
}
```

- A) 8
- B) 16
- C) 4
- D) 32

**Answer:** A) 8

**12. Which of the following problems is best solved using recursion?**

- A) Finding the largest element in an array
- B) Checking if a number is prime
- C) Tower of Hanoi
- D) Sorting an array



**Answer:** C) Tower of Hanoi

**13. What will be the output of this function?**

```
public class Main {  
    public static void print(int n) {  
        if (n == 0) return;  
        System.out.print(n + " ");  
        print(n - 1);  
        System.out.print(n + " ");  
    }  
    public static void main(String[] args) {  
        print(3);  
    }  
}
```

- A) 3 2 1 1 2 3
- B) 3 2 1
- C) 1 2 3
- D) 3 2 1 2 3

**Answer:** A) 3 2 1 1 2 3

**14. What is the space complexity of a recursive function that does not use extra variables?**

- A)  $O(1)$
- B)  $O(n)$
- C)  $O(\log n)$
- D)  $O(n^2)$

**Answer:** B)  $O(n)$

**15. Which of the following sorting algorithms is based on recursion?**

- A) Bubble Sort
- B) Quick Sort
- C) Selection Sort
- D) Insertion Sort

**Answer:** B) Quick Sort

---

**16. What will be the output of the following function?**

```
public class Main {  
    public static int power(int base, int exp) {  
        if (exp == 0) return 1;  
        return base * power(base, exp - 1);  
    }  
    public static void main(String[] args) {  
        System.out.println(power(2, 3));  
    }  
}
```

- A) 8
- B) 6
- C) 9
- D) 12

**Answer:** A) 8

**17. What is recursion in Java?**

- a) A function calling another function
- b) A function calling itself
- c) A function that runs infinitely
- d) A function with multiple parameters

✓ **Answer:** b) A function calling itself

**18. What is the base case in recursion?**

- a) The case when the function calls itself
- b) The condition that stops the recursion
- c) The middle element in recursion
- d) The first function call

✓ **Answer:** b) The condition that stops the recursion

**19. What happens if a recursive function has no base case?**

- a) It runs once and stops
- b) It runs indefinitely (StackOverflowError)
- c) It runs in a loop
- d) It gives a compilation error

✓ **Answer:** b) It runs indefinitely (StackOverflowError)

**20. What data structure is used for function calls in recursion?**

- a) Queue
- b) Stack
- c) Heap

d) Linked List

✓ Answer: b) Stack

**21. What is the primary advantage of recursion?**

a) It makes code easier to understand

b) It improves execution speed

c) It reduces memory usage

d) It always avoids loops

✓ Answer: a) It makes code easier to understand

**22. What is the time complexity of computing the nth Fibonacci number using recursion?**

a)  $O(n)$

b)  $O(\log n)$

c)  $O(2^n)$

d)  $O(n^2)$

✓ Answer: c)  $O(2^n)$

**23. Which of the following problems is best solved using recursion?**

a) Sorting an array

b) Tower of Hanoi

c) Printing elements of an array

d) Multiplication of two numbers

✓ Answer: b) Tower of Hanoi

**24. Which of the following is a recursive function?**

```
public int sum(int n) {  
    if (n == 0) return 0;  
    return n + sum(n - 1);  
}
```

a) Yes

b) No

✓ Answer: a) Yes

**25. What is the output of the following code?**

```
public static void recursive(int n) {  
    if (n == 0) return;  
    System.out.print(n + " ");  
    recursive(n - 1);  
}  
public static void main(String[] args) {  
    recursive(3);  
}
```

}

- a) 3 2 1 0
- b) 3 2 1
- c) 1 2 3
- d) Compilation error

✓ Answer: b) 3 2 1

**26. Which of the following is a tail recursive function?**

```
public int factorial(int n, int result) {  
    if (n == 0) return result;  
    return factorial(n - 1, n * result);  
}
```

- a) Yes
- b) No

✓ Answer: a) Yes

**27. What is tail recursion?**

- a) A recursive call made at the beginning of the function
- b) A recursive call made at the end of the function
- c) A recursive call inside a loop
- d) A recursive function with multiple parameters

✓ Answer: b) A recursive call made at the end of the function

**28. Which sorting algorithm uses recursion?**

- a) Bubble Sort
- b) Merge Sort
- c) Selection Sort
- d) Insertion Sort

✓ Answer: b) Merge Sort

**29. What is the worst-case time complexity of QuickSort using recursion?**

- a)  $O(n \log n)$
- b)  $O(n^2)$
- c)  $O(n)$
- d)  $O(\log n)$

✓ Answer: b)  $O(n^2)$

**30. What is the recurrence relation for the Fibonacci sequence?**

- a)  $F(n) = F(n-1) + F(n-2)$
- b)  $F(n) = F(n-1) * F(n-2)$
- c)  $F(n) = F(n-1) - F(n-2)$

d)  $F(n) = F(n-1) / F(n-2)$

✓ **Answer:** a)  $F(n) = F(n-1) + F(n-2)$

**31. What is the base case in the recursive factorial function?**

a)  $n == 1$

b)  $n == 0$

c)  $n == -1$

d) No base case

✓ **Answer:** b)  $n == 0$

**32. What will be printed by the following recursive function?**

```
public static void print(int n) {  
    if (n > 5) return;  
    System.out.print(n + " ");  
    print(n + 1);  
}  
public static void main(String[] args) {  
    print(1);  
}
```

a) 1 2 3 4 5

b) 1 2 3 4 5 6

c) Infinite recursion

d) Compilation error

✓ **Answer:** a) 1 2 3 4 5

**33. What will be the output?**

```
public static void mystery(int n) {  
    if (n == 0) return;  
    mystery(n - 1);  
    System.out.print(n + " ");  
}  
public static void main(String[] args) {  
    mystery(3);  
}
```

a) 1 2 3

b) 3 2 1

c) 0 1 2 3

d) Infinite loop

✓ **Answer:** a) 1 2 3

**34. How can recursion be optimized?**

- a) Using loops
- b) Using memoization
- c) Using switch-case
- d) Increasing function calls

✓ **Answer:** b) Using memoization

**35. Which problem does not use recursion?**

- a) Factorial calculation
- b) Fibonacci series
- c) Prime number check
- d) Palindrome check

✓ **Answer:** c) Prime number check

**36. Which of the following best describes indirect recursion?**

- a) A function calling itself
- b) Two or more functions calling each other in a cycle
- c) A function that never stops
- d) A function with multiple parameters

✓ **Answer:** b) Two or more functions calling each other in a cycle