JAVA – 100 HARD CODE-BASED MCQs

(WITH ANSWERS AT END)

🔹 TOPIC 1: JVM, Memory, Data Types, Operators (Q1–Q15)

Q1

```
class Test {
    public static void main(String[] args) {
        System.out.println(10 + 20 * 30);
    }
}
```

A) 900
B) 600
C) 610
D) 700

Q2

```
class Test {
    public static void main(String[] args) {
        System.out.println('A' + 1);
    }
}
```

A) A1
B) 66
C) Compilation error
D) Runtime error

Q3

```
class Test {
    public static void main(String[] args) {
        byte b = 10;
        b = b + 1;
        System.out.println(b);
    }
}
```

A) 11
B) 10
C) Compilation error
D) Runtime error

Q4

```
class Test {
    public static void main(String[] args) {
        System.out.println(10 == 10.0);
    }
}
```

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

Q5

```
class Test {
    public static void main(String[] args) {
        System.out.println(null == null);
    }
}
```

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

Q6

```
class Test {
    public static void main(String[] args) {
        int x = 5;
        System.out.println(x++ + ++x);
    }
}
```

A) 11
B) 12
C) 13
D) 14

Q7

```
class Test {
    public static void main(String[] args) {
        System.out.println(0.1 + 0.2 == 0.3);
    }
}
```

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

Q8

```
class Test {
    static int x;
    public static void main(String[] args) {
        System.out.println(x);
    }
}
```

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

Q9

```
class Test {
    public static void main(String[] args) {
        System.out.println(Math.min(Double.MIN_VALUE, 0.0d));
    }
}
```

A) Double.MIN_VALUE
B) 0.0
C) Compilation error
D) Runtime error

Q10

```
class Test {
    public static void main(String[] args) {
        int x = 10;
        x += 5;
        System.out.println(x);
```

```
    }
}
```

A) 10
B) 15
C) Compilation error
D) Runtime error

Q11

```
class Test {
    public static void main(String[] args) {
        System.out.println(5 / 2);
    }
}
```

A) 2
B) 2.5
C) Compilation error
D) Runtime error

Q12

```
class Test {
    public static void main(String[] args) {
        System.out.println(5 / 2.0);
    }
}
```

A) 2
B) 2.5
C) Compilation error
D) Runtime error

Q13

```
class Test {
    public static void main(String[] args) {
        int x = 10;
        System.out.println(x == 10 ? "Yes" : "No");
    }
}
```

A) true
B) false
C) Yes
D) No

Q14

```
class Test {
    public static void main(String[] args) {
        System.out.println(10 << 2);
    }
}
```

A) 20
B) 40
C) 80
D) 5

Q15

```
class Test {
    public static void main(String[] args) {
        System.out.println(-10 >>> 2);
    }
}
```

A) -2
B) -3
C) Large positive number
D) Compilation error

✅ ANSWERS (Q1–Q15)

1. C
2. B
3. C
4. A
5. A
6. C
7. B
8. B
9. B
10. B
11. A
12. B
13. C
14. B
15. C

TOPIC 2: Strings, StringBuilder, Wrapper Classes (Q16–Q30)

Q16

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

A) JavaWorld
B) Java
C) Compilation error
D) Runtime error

Q17

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

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

Q18

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

A) true
B) false

C) Compilation error
D) Runtime error

Q19

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

A) Java
B) JavaSE
C) Java SE
D) Compilation error

Q20

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

A) Java
B) World
C) JavaWorld
D) Compilation error

Q21

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

A) Java
B) avaJ
C) Compilation error
D) Runtime error

Q22

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

A) Java
B) nullJava
C) NullPointerException
D) Compilation error

Q23

```
class Test {
    public static void main(String[] args) {
        Integer i1 = 127;
        Integer i2 = 127;
        System.out.println(i1 == i2);
    }
}
```

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

Q24

```
class Test {
    public static void main(String[] args) {
        Integer i1 = 128;
        Integer i2 = 128;
        System.out.println(i1 == i2);
    }
}
```

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

Q25

```
class Test {
    public static void main(String[] args) {
        System.out.println(Integer.parseInt("10") + 20);
    }
}
```

A) 30
B) 1020
C) Compilation error
D) Runtime error

Q26

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

A) Ja
B) av
C) va
D) ava

Q27

```
class Test {
    public static void main(String[] args) {
        String s = "Java";
        System.out.println(s.charAt(4));
    }
}
```

A) a
B) Exception
C) Compilation error
D) Runtime error

Q28

```
class Test {
    public static void main(String[] args) {
        System.out.println("Java".equals(new String("Java")));
    }
}
```

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

Q29

```
class Test {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder();
        System.out.println(sb.capacity());
    }
}
```

A) 0
B) 10
C) 16
D) 32

Q30

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

A) JAVA
B) Java
C) Compilation error
D) Runtime error

✅ ANSWERS (Q16–Q30)

16. B
17. A
18. B
19. C
20. C
21. B
22. B
23. A
24. B

25. A
26. B
27. D
28. A
29. C
30. B

TOPIC 3: OOP, Inheritance, Polymorphism, `super`, `this` (Q31–Q50)

Q31

```
class A {
    A() {
        System.out.print("A");
    }
}
class B extends A {
    B() {
        System.out.print("B");
    }
}
public class Test {
    public static void main(String[] args) {
        new B();
    }
}
```

A) A
B) B
C) AB
D) BA

Q32

```
class A {
    void m1() {
        System.out.println("A");
    }
}
class B extends A {
    void m1() {
        System.out.println("B");
    }
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.m1();
    }
```

```
}
```

A) A
B) B
C) Compilation error
D) Runtime error

Q33

```
class A {
    static void m1() {
        System.out.println("A");
    }
}
class B extends A {
    static void m1() {
        System.out.println("B");
    }
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.m1();
    }
}
```

A) A
B) B
C) Compilation error
D) Runtime error

Q34

```
class A {
    final void m1() {
        System.out.println("A");
    }
}
class B extends A {
    void m1() {
        System.out.println("B");
    }
}
```

A) Compiles successfully
B) Compilation error
C) Runtime error
D) No output

Q35

```
class A {
    A() {
        this(10);
        System.out.print("A");
    }
    A(int x) {
        System.out.print(x);
    }
}
public class Test {
    public static void main(String[] args) {
        new A();
    }
}
```

A) A10
B) 10A
C) Compilation error
D) Runtime error

Q36

```
class A {
    int x = 10;
}
class B extends A {
    int x = 20;
    void print() {
        System.out.println(super.x);
    }
}
public class Test {
    public static void main(String[] args) {
        new B().print();
    }
}
```

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

Q37

```
class A {
    private void m1() {
        System.out.println("A");
    }
}
class B extends A {
```

```
    void m1() {
        System.out.println("B");
    }
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.m1();
    }
}
```

A) A
B) B
C) Compilation error
D) Runtime error

Q38

```
class A {
    int x = 10;
}
class B extends A {
    int x = 20;
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        System.out.println(obj.x);
    }
}
```

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

Q39

```
abstract class A {
    abstract void m1();
}
class B extends A {
    void m1() {
        System.out.println("B");
    }
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.m1();
    }
}
```

A) A
B) B
C) Compilation error
D) Runtime error

Q40

```
interface A {
    void m1();
}
class B implements A {
    public void m1() {
        System.out.println("B");
    }
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        obj.m1();
    }
}
```

A) A
B) B
C) Compilation error
D) Runtime error

Q41

```
class A {
    static {
        System.out.print("A");
    }
}
class B extends A {
    static {
        System.out.print("B");
    }
}
public class Test {
    public static void main(String[] args) {
        new B();
    }
}
```

A) A
B) B
C) AB
D) BA

Q42

```
class A {
    int x = 10;
}
class B extends A {
    B() {
        System.out.println(x);
    }
}
public class Test {
    public static void main(String[] args) {
        new B();
    }
}
```

A) 0
B) 10
C) Compilation error
D) Runtime error

Q43

```
class A {
    A() {
        System.out.print("A");
    }
}
class B extends A {
    B() {
        super();
        System.out.print("B");
    }
}
public class Test {
    public static void main(String[] args) {
        new B();
    }
}
```

A) A
B) B
C) AB
D) BA

Q44

```
class A {
    static int x = 10;
}
class B extends A {
    static int x = 20;
```

```
}
public class Test {
    public static void main(String[] args) {
        A obj = new B();
        System.out.println(obj.x);
    }
}
```

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

Q45

```
class A {
    A() {
        System.out.println("A");
    }
}
class B extends A {
    B() {
        System.out.println("B");
    }
}
class C extends B {
    C() {
        System.out.println("C");
    }
}
public class Test {
    public static void main(String[] args) {
        new C();
    }
}
```

A) C
B) BC
C) ABC
D) CBA

Q46

```
class A {
    void m1() {
        System.out.println("A");
    }
}
class B extends A {
    void m1() {
        System.out.println("B");
    }
```

```
    void call() {
        super.m1();
    }
}
public class Test {
    public static void main(String[] args) {
        new B().call();
    }
}
```

A) A
B) B
C) AB
D) BA

Q47

```
class A {
    static void m1() {
        System.out.println("A");
    }
}
class B extends A {
    void m1() {
        System.out.println("B");
    }
}
```

A) Compiles successfully
B) Compilation error
C) Runtime error
D) No output

Q48

```
class A {
    protected void m1() {}
}
class B extends A {
    void m1() {}
}
```

A) Compiles successfully
B) Compilation error
C) Runtime error
D) No output

Q49

```
class A {
    public static void main(String[] args) {
        System.out.println("A");
    }
}
class B extends A {}
```

Running `java B` gives:
A) A
B) Compilation error
C) Runtime error
D) No output

Q50

```
class Test {
    Test() {
        System.out.println("Default");
    }
    Test(int x) {
        this();
        System.out.println(x);
    }
    public static void main(String[] args) {
        new Test(5);
    }
}
```

A) 5
B) Default
C) Default 5
D) Compilation error

✅ ANSWERS (Q31–Q50)

31. C
32. B
33. A
34. B
35. B
36. A
37. C
38. A
39. B
40. B
41. C
42. B
43. C

44. A
45. C
46. A
47. B
48. B
49. A
50. C

🔹 TOPIC 4: Exception Handling & Multithreading (Q51–Q70)

Q51

```
class Test {
    public static void main(String[] args) {
        try {
            System.out.println(10 / 0);
        } catch (ArithmeticException e) {
            System.out.println("AE");
        } finally {
            System.out.println("Finally");
        }
    }
}
```

A) AE
B) Finally
C) AE Finally
D) Compilation error

Q52

```
class Test {
    static void m1() throws Exception {
        throw new Exception();
    }
    public static void main(String[] args) {
        try {
            m1();
        } catch (RuntimeException e) {
            System.out.println("RE");
        } catch (Exception e) {
            System.out.println("E");
        }
    }
}
```

A) RE
B) E

C) Compilation error
D) Runtime error

Q53

```
class Test {
    public static void main(String[] args) {
        try {
            return;
        } finally {
            System.out.println("Finally");
        }
    }
}
```

A) No output
B) Finally
C) Compilation error
D) Runtime error

Q54

```
class Test {
    public static void main(String[] args) {
        try {
            System.out.println("Try");
            System.exit(0);
        } finally {
            System.out.println("Finally");
        }
    }
}
```

A) Try
B) Finally
C) Try Finally
D) No output

Q55

```
class Test {
    public static void main(String[] args) {
        try {
            System.out.println("Try");
        }
    }
}
```

A) Try
B) Compilation error
C) Runtime error
D) No output

Q56

```
class Test extends Thread {
    public void run() {
        System.out.println("Run");
    }
    public static void main(String[] args) {
        Test t = new Test();
        t.start();
        t.start();
    }
}
```

A) Run Run
B) Run
C) Runtime exception
D) Compilation error

Q57

```
class Test implements Runnable {
    public void run() {
        System.out.println("Run");
    }
    public static void main(String[] args) {
        Thread t = new Thread(new Test());
        t.run();
    }
}
```

A) Run in new thread
B) Run in main thread
C) No output
D) Compilation error

Q58

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread();
        t.start();
        System.out.println(t.getState());
    }
```

```
}
```

A) NEW
B) RUNNABLE
C) TERMINATED
D) BLOCKED

Q59

```
class Test extends Thread {
    public void run() {
        System.out.println(Thread.currentThread().getName());
    }
    public static void main(String[] args) {
        new Test().start();
    }
}
```

A) main
B) Thread-0
C) Thread-1
D) Compilation error

Q60

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread(() -> System.out.println("Run"));
        t.start();
    }
}
```

A) Run
B) Compilation error
C) Runtime error
D) No output

Q61

```
class Test {
    public static synchronized void m1() {
        System.out.println("Static");
    }
    public synchronized void m2() {
        System.out.println("Instance");
    }
}
```

Which lock is used by `m1()`?
A) Object lock
B) Class-level lock
C) No lock
D) Both

Q62

```
class Test extends Thread {
    public void run() {
        System.out.println("Run");
    }
}
public class Demo {
    public static void main(String[] args) {
        Test t = new Test();
        t.run();
    }
}
```

A) New thread
B) Main thread
C) No output
D) Compilation error

Q63

```
class Test {
    public static void main(String[] args) throws Exception {
        Thread.sleep(1000);
        System.out.println("Awake");
    }
}
```

A) Awake immediately
B) Awake after 1 sec
C) Compilation error
D) Runtime error

Q64

```
class Test {
    public static void main(String[] args) {
        System.out.println(Thread.currentThread().getPriority());
    }
}
```

Default priority?
A) 1
B) 5
C) 10
D) OS dependent

Q65

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread();
        System.out.println(t.isAlive());
    }
}
```

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

Q66

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread();
        t.setDaemon(true);
        t.start();
        System.out.println(t.isDaemon());
    }
}
```

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

Q67

```
class Test {
    public static void main(String[] args) {
        synchronized(Test.class) {
            System.out.println("Lock");
        }
    }
}
```

A) Object lock
B) Class lock
C) No lock
D) Runtime error

Q68

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread();
        t.start();
        t.join();
        System.out.println("End");
    }
}
```

A) Compilation error
B) Runtime error
C) End
D) No output

Q69

```
class Test {
    public static void main(String[] args) {
        System.out.println(Thread.holdsLock(new Object()));
    }
}
```

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

Q70

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread();
        t.start();
        t.start();
    }
}
```

A) No output
B) Runtime exception

C) Compilation error
D) Infinite loop

✅ ANSWERS (Q51–Q70)

51. C
52. B
53. B
54. A
55. B
56. C
57. B
58. B
59. B
60. A
61. B
62. B
63. B
64. B
65. B
66. A
67. B
68. C
69. B
70. B

TOPIC 5: Collections, Generics, File I/O, Miscellaneous (Q71–Q100)

Q71

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        List list = new ArrayList();
        list.add("Java");
        list.add(10);
        System.out.println(list);
    }
}
```

A) Compilation error
B) Runtime error
C) [Java, 10]
D) [10, Java]

Q72

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("Java");
        list.add(null);
        System.out.println(list);
    }
}
```

A) Compilation error
B) Runtime error
C) [Java, null]
D) [null, Java]

Q73

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        Set<Integer> set = new HashSet<>();
        set.add(10);
        set.add(10);
        System.out.println(set.size());
    }
}
```

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

Q74

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        Map<String, Integer> map = new HashMap<>();
        map.put("A", 1);
        map.put("A", 2);
        System.out.println(map.get("A"));
    }
}
```

A) 1
B) 2

C) null
D) Compilation error

Q75

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        List<Integer> list = Arrays.asList(1, 2, 3);
        list.add(4);
        System.out.println(list);
    }
}
```

A) [1, 2, 3, 4]
B) Compilation error
C) Runtime exception
D) [1, 2, 3]

Q76

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
        list.add(1);
        list.add(2);
        list.add(3);
        list.remove(1);
        System.out.println(list);
    }
}
```

A) [1, 2, 3]
B) [1, 3]
C) [2, 3]
D) [1, 2]

Q77

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        Queue<Integer> q = new PriorityQueue<>();
        q.add(3);
        q.add(1);
        q.add(2);
        System.out.println(q.poll());
    }
```

```
}
```

A) 3
B) 2
C) 1
D) Order not defined

Q78

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        TreeSet<Integer> set = new TreeSet<>();
        set.add(null);
        System.out.println(set);
    }
}
```

A) [null]
B) Compilation error
C) Runtime exception
D) []

Q79

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        Map<Integer, String> map = new Hashtable<>();
        map.put(1, null);
        System.out.println(map);
    }
}
```

A) {1=null}
B) {}
C) Compilation error
D) Runtime exception

Q80

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        List<String> list = new Vector<>();
        list.add("A");
        list.add("B");
        System.out.println(list);
```

```
    }
}
```

A) Compilation error
B) Runtime error
C) [A, B]
D) {}

Q81

```
import java.util.*;
class Test {
    public static void main(String[] args) {
        Iterator<Integer> it = new ArrayList<Integer>().iterator();
        System.out.println(it.hasNext());
    }
}
```

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

Q82

```
class Test<T> {
    T obj;
    Test(T obj) {
        this.obj = obj;
    }
    T get() {
        return obj;
    }
}
public class Demo {
    public static void main(String[] args) {
        Test<Integer> t = new Test<>(10);
        System.out.println(t.get());
    }
}
```

A) Compilation error
B) Runtime error
C) 10
D) null

Q83

```
import java.io.*;
class Test {
    public static void main(String[] args) throws Exception {
        File f = new File("abc.txt");
        System.out.println(f.exists());
    }
}
```

A) true always
B) false always
C) Depends if file exists
D) Compilation error

Q84

```
import java.io.*;
class Test {
    public static void main(String[] args) throws Exception {
        File f = new File("abc.txt");
        f.createNewFile();
        System.out.println(f.exists());
    }
}
```

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

Q85

```
import java.io.*;
class Test {
    public static void main(String[] args) throws Exception {
        FileOutputStream fos = new FileOutputStream("a.txt");
        fos.write(65);
        fos.close();
    }
}
```

What is written to file?
A) 65
B) 'A'
C) Compilation error
D) Runtime error

Q86

```
class Test {
    public static void main(String[] args) {
        System.out.println(String.format("%d", 10.5));
    }
}
```

A) 10
B) 10.5
C) IllegalFormatConversionException
D) Compilation error

Q87

```
enum Day {
    MON, TUE;
}
class Test {
    public static void main(String[] args) {
        System.out.println(Day.MON.ordinal());
    }
}
```

A) 0
B) 1
C) Compilation error
D) Runtime error

Q88

```
class Test {
    public static void main(String[] args) {
        assert false : "Error";
        System.out.println("Done");
    }
}
```

When assertions are enabled:
A) Done
B) Error
C) AssertionError
D) Compilation error

Q89

```
class Test {
    public static void main(String[] args) {
        System.out.println(Class.forName("java.lang.String"));
    }
```

```
}
```

A) java.lang.String
B) class java.lang.String
C) Compilation error
D) Runtime error

Q90

```
class Test {
    public static void main(String[] args) {
        System.out.println(Void.TYPE);
    }
}
```

A) void
B) Void
C) null
D) Compilation error

Q91

```
class Test {
    static {
        System.out.println("Static");
    }
    public static void main(String[] args) {
        System.out.println("Main");
    }
}
```

A) Main
B) Static
C) Static Main
D) Compilation error

Q92

```
class Test {
    public static void main(String[] args) {
        System.out.println(Math.round(10.5));
    }
}
```

A) 10
B) 11

C) 10.5
D) Compilation error

Q93

```
class Test {
    public static void main(String[] args) {
        System.out.println("Java".compareTo("Java"));
    }
}
```

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

Q94

```
class Test {
    public static void main(String[] args) {
        System.out.println(System.getProperty("java.version"));
    }
}
```

A) Compilation error
B) Runtime error
C) JVM version
D) null

Q95

```
class Test {
    public static void main(String[] args) {
        System.out.println(Runtime.getRuntime().availableProcessors());
    }
}
```

A) Always 1
B) Number of CPU cores
C) JVM version
D) Compilation error

Q96

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

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

Q97

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

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

Q98

```
class Test {
    public static void main(String[] args) {
        System.out.println(Boolean.valueOf("true"));
    }
}
```

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

Q99

```
class Test {
    public static void main(String[] args) {
        System.out.println(Character.isDigit('5'));
    }
}
```

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

Q100

```
class Test {
    public static void main(String[] args) {
        System.out.println(Math.max('A', 100));
    }
}
```

A) A
B) 65
C) 100
D) Compilation error

✅ ANSWERS (Q71–Q100)

71. C
72. C
73. B
74. B
75. C
76. B
77. C
78. C
79. D
80. C
81. B
82. C
83. C
84. B
85. B
86. C
87. A
88. C
89. B
90. A
91. C
92. B
93. C
94. C
95. B
96. A

97. B
98. A
99. A
100. C