

What will be the result of compiling and executing Test class?

```
public class Test {
    public static void main(String[] args) {
        int a = 7; // a = 7, 8, 9
        boolean res = a++ == 7 && ++a == 9 || a++ == 9; // 7==7(true) && 9==9(true)
    }
}
|| a++==9 = true
    System.out.println("a = " + a);
    System.out.println("res = " + res);
}
```

A. a=10  
res=true  
B. a=9  
res=true  
C. a=10  
res=false  
D. compilation error

Answer: B

```
Q>
class Fork {
    public static void main(String[] args) {
        if(args.length == 1 | args[1].equals("test"))
            System.out.println("test case");
        } else {
            System.out.println("production " + args[0]);
        }
    }
}
```

And the command-line invocation:

```
java Fork live2
```

What is the result?

- A. test case
- B. production live2
- C. test case live2
- D. Compilation fails
- E. An exception is thrown at runtime

JVM

```
args[0] = live2
```

```
args.length= 1
```

Answer: E(ArrayIndexOutOfBoundsException becoz args[1] is not available)

```
Q>
public class Test {
    public static void main(String[] args) {
        int aVar = 9; // aVar = 10
        if(aVar++ < 10) // if(9<10)true
            System.out.println(aVar + " Hello World!");
        else
            System.out.println(aVar + " Hello Universe!");
    }
}
```

A. 10 Hello World!  
B. 9 Hello Universe!  
C. 10 Hello World

D. Compilation fails.

Answer: A

```
Q>
public class Test {
    public static void main(String[] args) {
        int[] a = {1,2,3,4,5};
        for (xxx)
        {
            System.out.print(a[e]);
        }
    }
}
```

Which option can replace xxx to enable the code to print 135?

- A. int e = 0; e <= 4; e++
- B. int e = 0; e < 5; e += 2
- C. int e = 1; e <= 5; e += 1
- D. int e = 1; e < 5; e += 2

```
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
a[4] = 5
```

Answer: B(becoz it should print data present in even index only)

```
Q>
public class Test {
    public static void main(String[] args) {
        int num = 5; // num=5,4
        do{
            System.out.println(num-- + ""); // System.out.println(5)
        }while (num == 0);
        4==0(false)
    }
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed
- F. 4
- G. CompileTimeError

Answer: D

```
Q>
public class Test {
    public static void main(String[] args) {
        int ii=0;
        int jj=7;
        for (ii=0; ii<jj-1; ii=ii+2 ){
            System.out.print(ii);
        }
    }
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

JVM

```
ii =0 ,0<6(true)
ii=2, 2<6(true)
ii=4,4<6(true)
ii=6, 6<6(false)
```

Answer: 0 2 4

Q>

```
public class Test {
    public static void main(String[] args) {
        //line-3
        array[0] = 10;
        array[1] = 20;
        System.out.print(array[0]+":" array[1]);
    }
}
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

- A. `int[] array n= new int[2];`
- B. `int[] array;`  
`array = new int[2];`
- C. `int array = new int[2];`
- D. `int array [2] ;`

JVM

- A. `int[] array n= new int[2];`//invalid identifier
- B. `int[] array;`//array declaration  
`array = new int[2];`//array initialization
- C. `int array = new int[2];`//[] is missing
- D. `int array [2] ;`//new is missing

Q>

```
public class Test {
    public static void main(String[] args) {
        boolean opt =true;
        switch (opt){
            case true: System.out.print("True");
                       break;
            default : System.out.print("****");
        }
        System.out.print("Done");
    }
}
```

Predict the Output?

- A. CompileTime Error.
- B. Some problem occurred by jvm during execution.
- C. True
- D. True

\*\*\*  
Done

E. None of the above.

Note:

```
switch(arg)
    |-> byte---> Byte
        |-> short--->Short
        |-> int    ---> Integer
        |-> char   ---> Character
    |->String
    |-> enum
```

Answer: A

Q>

```
public class Test {
    public static void main(String[] args) {
        int intArr[] = {15,30,45,60,75};
        intArr[2] =intArr[4];
        intArr[4] = 90;
    }
}
```

What are the values of each element in intArr after this code has executed?

- A. 15, 60, 45, 90, 75
- B. 15, 90, 45, 90, 75
- C. 15, 30, 75, 60, 90
- D. 15, 30, 90, 60, 90
- E. 15, 4, 45, 60, 90

```
intArr[0] = 15
intArr[1] = 30
intArr[2] = 45,75
intArr[3] = 60
intArr[4] = 75,90
```

Output: 15,30,75,60,90

Answer: C

eg#3.

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

- A. CE
- B. some problem at runtime
- C. hello
- D. none of the above

Answer: C (becoz x is not used)

eg#4

```
class Test{
    public static void main(String[] args){
        int x;//local variable ---> no default initialization by jvm,
        programmer should do
        System.out.println(x);
    }
}
```

- A. CE
- B. some problem at runtime
- C. hello
- D. 0
- E. none of the above

Answer: A (becoz x is not initialized)

eg#5.

```
class Test{
    public static void main(String[] args) {
        int x;
        if(args.length>0){
            x=10;
        }
        System.out.println(x);
    }
}
```

Assume java Test 10 is passed throug command line

- A. CE
- B. some problem at runtime
- C. hello
- D. 10
- E. none of the above

Answer: A(x = 10 is initialized inside block ,no gaurantee that block would be executed by compiler)

eg#6

```
class Test {
    public static void main(String[] args) {
        int x;
        if(args.length>0){
            x=10;
        }else{
            x=20;
        }
        System.out.println(x);
    }
}
```

Assume java Test sachin is passed throug command line

- A. CE
- B. some problem at runtime
- C. hello
- D. 10
- E. 20
- F. None of the above

```
java Test sachin
args[0] ="sachin"
args.length = 1
x =10
```

```
java Test
args.length = 0
x =20
```



