

1. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");  
2 Person p2 = new Person("Jiyoon");  
3 System.out.println(p1 != p2);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

`true`

- E. One line output to the console:

`false`

- F. None of the above.

2. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");  
2 Person p2 = new Person("Jiyoon");  
3 Person[] persons = new Person[2];  
4 System.out.println(persons[persons.length()] != null);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

`true`

- E. One line output to the console:

`false`

- F. None of the above.

3. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = new Person[2];
4 System.out.println(persons[persons.length] != null);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

true

- E. One line output to the console:

false

- F. None of the above.

4. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = new Person[2];
4 System.out.println(persons[persons.length - 1] != null);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

true

- E. One line output to the console:

false

- F. None of the above.

5. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = new Person[2];
4 System.out.println(persons[persons.length - 1].name.equals("Jiyoon"));
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.**
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

true

- E. One line output to the console:

false

- F. None of the above.

6. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = {p1, p2};
4 p1 = p2;
5 System.out.println(persons[0] == p1);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

true

- E. One line output to the console:

false

- F. None of the above.

7. Assume that a `Person` class is already defined, and it has an attribute `name` and a constructor that initializes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = {p1, p2};
4 p1 = p2;
5 persons[0] = p2;
6 System.out.println(persons[0] == p1);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

true

- E. One line output to the console:

false

- F. None of the above.

8. Assume that a `Person` class is already defined, and it has an attribute `name`, a constructor that initializes the person's name from the input string, and a mutator method `setName` that changes the person's name from the input string. Consider the following fragment of Java code (inside some `main` method):

```
1 Person p1 = new Person("Heeyeon");
2 Person p2 = new Person("Jiyoon");
3 Person[] persons = {p1, p2};
4 p1 = persons[1];
5 persons[0] = p2;
6 p2.setName("Jihye");
7 System.out.println(p1.name);
```

What happens when executing the above Java code?

- A. The above Java code does not compile.
- B. A `NullPointerException` occurs.
- C. An `ArrayIndexOutOfBoundsException` occurs.
- D. One line output to the console:

Heeyeon

- E. One line output to the console:

Jiyoon

- F. One line output to the console:

Jihye

- G. None of the above.

9. Consider the following fragment of Java code:

```
int[][] sa = {
    {1, 8, 9},
    {2, 6, 7, 23},
    {3, 2, 5, 2, 1}
};
int sumOfRow = 0;
for(int r = 0; r < sa.length; r++) {
    int maxSum = 0;
    int rowWithMaxSum = 0;
    for(int c = 0; c < sa[r].length; c++) {
        sumOfRow += sa[r][c];
        if(sumOfRow > maxSum) {
            rowWithMaxSum = r;
            maxSum = sumOfRow;
        }
    }
    System.out.println("Sum of row " + r + ": " + sumOfRow);
    System.out.println("Row with max sum: " + rowWithMaxSum);
    System.out.println("Max sum: " + maxSum);
}
```

Write down the precise output to the console from the above program.

Solution:

```
Sum of row 0: 18
Row with max sum: 0
Max sum: 18
Sum of row 1: 33
Row with max sum: 1
Max sum: 33
Sum of row 2: 43
Row with max sum: 2
Max sum: 43
```

[of 10 marks]

10. Given an 2-dimensional integer array whose name is a

```
int[][] a = ... /* initialized with some values */
```

Write, in valid Java syntax, a fragment of Java code that calculates and outputs: **1)** the product of sums of a's rows; and **2)** the sum of products of a's rows. For example, if a is

```
{ {1, 2, 3}, /* sum of row 0 is 6, product of row 0 is 6 */  
  {4, 5}, /* sum of row 1 is 9, product of row 1 is 20 */  
  {6} /* sum of row 2 is 6, product of row 2 is 6 */  
}
```

Then the your program should output:

```
Product of sums is 324  
Sum of products is 32
```

Solution:

```
int pOfS = 1;  
int sOfP = 0;  
for(int r = 0; r < a.length; r++) {  
    int sum = 0;  
    int product = 1;  
    for(int c = 0; c < a[r].length; c++) {  
        sum += a[r][c];  
        product *= a[r][c];  
    }  
    pOfS *= sum;  
    sOfP += product;  
}  
System.out.println("Product of sum is " + pOfS);  
System.out.println("Sum of product is " + sOfP);
```

[of 10 marks]