JetDevs

Java Test Instructions:

Code submission -
1. Create and submit the code to Github

2. Record demo - Upon completion, record a video of your desktop which shows the compilation of the test task and walk through a demo of your solutions. Also talk through the approach you took and what aspects of the problem were challenging to solve.

Time limit - Submit the project 24 hours upon receiving this test

Task requirements - Create a service with Spring boot, then write a SEPARATE interface for each of your solutions. You will be choosing five (5) problems to solve from the three groups below.

GROUP A (CHOOSE 2)

1. Remove invalid brackets from strings

```
API Path: /brackets
Method: Post
Response Body:
{
        text: “test (t(e)x)t)”
}
Response Body:
{
        text: “test (t(e)x)t”
}
```

2. Product availability problem
A product is available in multiple countries around the world and is only available from 09:00 -18:00 local time. Enter the time zone to determine if the service is available during local time.

3. Counting steps
Climb stairs, either 1 or 2 steps at a time, enter the number of steps and find the number of different steps, using recursion and no member variables.

4. Number calling function
Create a Java Demo: A bank has 3 consultation windows, 10 users. Use a queue to implement the number calling function.

GROUP B (CHOOSE 2)

5. Check if a single linked table is a circular linked table.

API Path: /linked
Method: Post
Request Body:

```
[
    { ParentPos: -1, Value: 1 } , // ParentPos:-1 - Indicates the head of the chain table.
    { ParentPos:  0, Value: 2 } , // ParentPos: 0 - Indicates the position of the parent node at 0 in the array.
    { ParentPos:  1, Value: 3 } , // ParentPos: 1 - Indicates the position of the parent node at 1 in the array.
    . . .
]
```

Response Body:

```
{
    looped: true|false,
    looped_item: null|{ Pos: ?, Value: ? }
}
```

6. Use nested conditional operators to solve this problem:

Students with academic scores >= 90 are represented by A, those with scores between 60 and 89 are represented by B, and those with scores below 60 are represented by C.

API Path: /students
Method: Post
Request Body:

```
[ 99, 79, 30, 66, … ]
```

Response Body:

```
[ A, B, C, B, … ]
```

7. Find the minimum number of platforms in a train station

This question will give the arrival and departure times of trains to a particular station. It is necessary to find the minimum number of platforms required at a station at any point in time.

API Path: /trains
Method: Post
Response Body:

```
{
    arrival: {“1:00”, “1:40”, “1:50”, “2:00”, “2:15, “4:00”},
    departure: {“1:10”, “3:00”, “2:20”, “2:30”, “3:15”, “6:00”}
}
```

Response Body:

```
{
    platforms: 4
}
```

8. Encryption
A says "HI" to B, and B responds with "Hello".

Simulate the HTTPS encryption and decryption process, requiring multiple encryption methods to ensure that Hi and Hello are not intercepted and deciphered.

GROUP C (CHOOSE 1)

9. Create a Java DEMO, implementing a deadlock.

10. Implement an object caching pool

```
API Path: /pool
Method: Post // Adding objects to the cache pool
Request Body:
[
        {
                key: “object 1”,
                value: 0
        },
        {
                key: “object 2”,
                value: 2
        },
        {
                key: “object 3”,
                value: 3
        },
]
Response Body:
{
        total_objects: 3 // Total number of objects in the current cache pool
}

Method: Get // Fetching objects from the cache pool
Query Parameter: /pool/{key}
Response Body:
Success:
{
        value: 3
}
```

Does not exist returns 404

Method: Delete // *Deleting objects from the cache pool*
Query Parameter: /pool/{key}
Response Body:
Success returns 200, non-existent returns 404