Test B

Q1,

```
class java {
	public static String reverse(String input) {
		char reversed[] = new char[input.length()];
		for (int i = 0; i < input.length(); i++) {
			reversed[i] = input.charAt(input.length() - 1 - i);
		}
		return new String(reversed);
	}

	public static void main(String[] args) {
		String input = "retlaohS";
		String output = reverse(input);
		System.out.println("Input: "+input);
		System.out.println("Output: "+output);
	}
}
```

Q2,

```
SELECT user_code, COUNT(*) AS packed_totes_count
FROM order_tote_process_log
WHERE action_code = 'PACKED'
AND process_date >= '2023-11-01 10:00:00'
AND process_date <= '2023-11-01 10:59:59'
GROUP BY user_code;
```

Expected output:

user_code	packed_totes_count
P1	1

Q3,

```
class java {
	public static int calculateMaxProductQuantity(double boxLength, double boxWidth, double boxHeight,
double productLength, double productWidth, double productHeight) {

		double lengthFit = boxLength /    productLength;
		double widthFit = (boxWidth / productWidth);
		double heightFit = (boxHeight /    productHeight);

		return (int)(lengthFit * widthFit * heightFit);
	}
```

```
    public static void main(String[] args) {
        double cartonLength = 320;
        double cartonWidth = 260;
        double cartonHeight = 200;
        double productLength = 210;
        double productWidth = 35;
        double productHeight = 35;

        int maxQuantity = calculateMaxProductQuantity(cartonLength, cartonWidth,
cartonHeight,productLength, productWidth, productHeight);
        System.out.println("maximum quantity of a product in carton box: " + maxQuantity);
    }
}
```

Q4,

1. API Design and Specification for shipping fee calculation

   Endpoint: /shipping/calculate

   Method: POST

   Request Body:

```
{
  "items": [
    {
      "length": "float, length of the item in cm",
      "width": "float, width of the item in cm",
      "height": "float, height of the item in cm",
      "weight": "float, weight of the item in kg",
      "quantity": "int, number of identical items"
    } ],
  "temperatureCondition": "string, either 'Ambient' or 'Chill'"
}
```

   Response:

```
{
  "totalShippingFee": "float, calculated total shipping fee in HKD",
  "itemFees": [
    {
      "itemIndex": "int, index of the item in the request",
      "chargedWeight": "float, higher of volumetric or physical weight in kg",
      "shippingFee": "float, calculated shipping fee for the item in HKD"
    }
  ]
}
```

Status Codes:

200 OK: The request was successful, and the response body contains the calculated fees.

400 Bad Request: The request was invalid. Details of the invalid fields should be provided in the response body.

500 Internal Server Error: An unexpected error occurred on the server.

2. Code in the folder.

Q5