Rules:

- Waktu pengerjaan 3 hari kerja, submit yang dikerjakan, selesai / tidak selesai, yang kami lihat adalah konsep, logic, kerapian dan pemahaman.
- Semua kodingan harus di push ke dalam Git, sertakan link git.
- Semakin sering melakukan commit, semakin baik.
- Sertakan Readme untuk soal aplikasi.
- Jika ada pertanyaan, silakan reply all email dari mcpayment.
- Gunakan penamaan variable yang baik dan benar, ikuti konsep SOLID adalah nilai plus.

1. Validating Sudoku

A **Sudoku board** can be represented as a 9x9 matrix. It is valid if the following three conditions are met:

- Each row contains unique values from 1-9.
- Each column contains unique values from 1-9.
- Each of the 9 sub-squares, of size 3x3, contains a unique value from 1-9.

Input Format

- The first line denotes the number of test cases.
- The following, every 9 lines consist of 9 space-separated integers each, representing a 9x9 Sudoku grid.

Output Format

- A single word as output for every test case: - Valid, if it is a valid sudoku solution. - Invalid, if it is not a valid sudoku solution.

Sample Input:

```
2
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
```

```
3 4 5 2 8 6 1 7 9
2 8 6 9 4 5 1 7 3
7 1 4 6 3 2 9 5 8
9 3 5 7 8 1 4 2 6
4 2 7 3 5 6 8 1 9
6 5 8 1 9 7 3 4 2
1 9 3 4 2 8 7 6 5
3 6 1 5 7 9 2 8 4
5 4 2 8 1 3 6 9 7
8 7 9 2 6 4 5 3 1
```

Sample Output:

```
Valid
Valid
```

2. Personal Finance

Create a simple web service to track balance, income and expenses.

- Database: SQLite.
- Decide yourself what endpoints and tables are needed.
- Add Test Cases for each endpoint.
- Add API documentation (Swagger) (Optional).
- Frameworks (Express, NestJS).
- Give all your best!