

# Question for candidate



Nutchanon

Modified Today

## Background

We have implemented an order placement system for producing custom films for mobile phone's screen. Customers place orders through various online platforms such as Shopee, Lazada, TikTok, Facebook, and Amazon. Internally, we use a specific product code to communicate with our production teams. However, due to the different product format requirements on each online platform, we need to implement logic to map these platform-specific product IDs to our internal codes for accurate production.

## Internal Product Code Format

Our internal product code follows this pattern:

**{film type ID} - {texture ID} - {phone model ID}**

### Examples:

- FG0A-CLEAR-IPHONE16PROMAX
- FG05-MATTE-OPPOA3
- FG0A-PRIVACY-SAMSUNGS25

However, for marketing or promotional purposes, product IDs on different platforms may appear in various formats, such as:

- **FG0A-CLEAR-IPHONE16PROMAX\*2**
- **FG0A-CLEAR-IPHONE16PROMAX/FG0A-CLEAR-IPHONE16PROMAX-B**
- **--FG0A-CLEAR-IPHONE16PROMAX**
- **%20-FG0A-CLEAR-IPHONE16PROMAX**

Additionally, we have conditions where we include complementary items, such as wiping cloths per ordered quantity and cleaners, based on the film's texture per ordered quantity.

To simplify the production process, we want to extract **film type ID** and **texture ID** to field name "materialId" and **phone model ID** into a field name "modelId" for better processing and efficiency.

```

1  //Example of struct
2
3  type InputOrder struct {
4      No          int    `json:"no"`
5      PlatformProductId string `json:"platformProductId"`
6      Qty          int    `json:"qty"`
7      UnitPrice    float64 `json:"unitPrice"`
8      TotalPrice   float64 `json:"totalPrice"`
9  }
10
11 type CleanedOrder struct {
12     No          int    `json:"no"`
13     ProductId   string `json:"productId"`
14     MaterialId  string `json:"materialId"`
15     ModelId     string `json:"modelId"`
16     Qty          int    `json:"qty"`
17     UnitPrice    float64 `json:"unitPrice"`
18     TotalPrice   float64 `json:"totalPrice"`
19 }
20
21

```

---

## We expect you to

- Implement service that at least can handle 7 cases of the section below **with the aim of high performance, maintainable and extensible**
- Write unit test to prove that your function is working as expect

The section below shows some cases of input and expectation after we clean it. The ordering of complementary items does not matter but should be behind the main item

## Case 1 : Only one product

### Input

```
1  [  
2    {  
3      "no": 1,  
4      "platformProductId": "FG0A-CLEAR-IPHONE16PROMAX",  
5      "qty": 2,  
6      "unitPrice": 50,  
7      "totalPrice": 100  
8    }  
9  ]
```

### Expected Output

```
1  [  
2    {  
3      "no": 1,  
4      "productId": "FG0A-CLEAR-IPHONE16PROMAX",  
5      "materialId": "FG0A-CLEAR",  
6      "modelId": "IPHONE16PROMAX",  
7      "qty": 2,  
8      "unitPrice": 50.00,  
9      "totalPrice": 100.00  
10   },  
11   {  
12     "no": 2,  
13     "productId": "WIPING-CLOTH",  
14     "qty": 2,  
15     "unitPrice": 0.00,  
16     "totalPrice": 0.00  
17   },  
18   {  
19     "no": 3,  
20     "productId": "CLEAR-CLEANNER",  
21     "qty": 2,  
22     "unitPrice": 0.00,  
23     "totalPrice": 0.00  
24   }  
25  ]
```

---

## Case 2: One product with wrong prefix

### Input

```
1  [  
2    {  
3      "no": 1,  
4      "platformProductId": "x2-3&FG0A-CLEAR-IPHONE16PROMAX",  
5      "qty": 2,  
6      "unitPrice": 50,  
7      "totalPrice": 100  
8    }  
9  ]
```

### Expected Output

```
1  [  
2    {  
3      "no": 1,  
4      "productId": "FG0A-CLEAR-IPHONE16PROMAX",  
5      "materialId": "FG0A-CLEAR",  
6      "modelId": "IPHONE16PROMAX",  
7      "qty": 2,  
8      "unitPrice": 50.00,  
9      "totalPrice": 100.00  
10   },  
11   {  
12     "no": 2,  
13     "productId": "WIPING-CLOTH",  
14     "qty": 2,  
15     "unitPrice": 0.00,  
16     "totalPrice": 0.00  
17   },  
18   {  
19     "no": 3,  
20     "productId": "CLEAR-CLEANNER",  
21     "qty": 2,  
22     "unitPrice": 0.00,  
23     "totalPrice": 0.00  
24   }  
25  ]
```

### Case 3: One product with wrong prefix and has \* symbol that indicates the quantity

#### Input

```
1  [  
2    {  
3      "no": 1,  
4      "platformProductId": "x2-3&FG0A-MATTE-IPHONE16PROMAX*3",  
5      "qty": 1,  
6      "unitPrice": 90,  
7      "totalPrice": 90  
8    }  
9  ]
```

#### Expected Output

```
1  [  
2    {  
3      "no": 1,  
4      "productId": "FG0A-MATTE-IPHONE16PROMAX",  
5      "materialId": "FG0A-MATTE",  
6      "modelId": "IPHONE16PROMAX",  
7      "qty": 3,  
8      "unitPrice": 30.00,  
9      "totalPrice": 90.00  
10   },  
11   {  
12     "no": 2,  
13     "productId": "WIPING-CLOTH",  
14     "qty": 3,  
15     "unitPrice": 0.00,  
16     "totalPrice": 0.00  
17   },  
18   {  
19     "no": 3,  
20     "productId": "MATTE-CLEANER",  
21     "qty": 3,  
22     "unitPrice": 0.00,  
23     "totalPrice": 0.00  
24   }  
25  ]
```

## Case 4: One bundle product with wrong prefix and split by / symbol into two product

### Input

```
1  [  
2    {  
3      "no": 1,  
4      "platformProductId": "FG0A-CLEAR-OPPOA3/%20xFG0A-CLEAR-OPPOA3-B",  
5      "qty": 1,  
6      "unitPrice": 80,  
7      "totalPrice": 80  
8    }  
9  ]
```

### Expected Output

```
1  [  
2    {  
3      "no": 1,  
4      "productId": "FG0A-CLEAR-OPPOA3",  
5      "materialId": "FG0A-CLEAR",  
6      "modelId": "OPPOA3",  
7      "qty": 1,  
8      "unitPrice": 40.00,  
9      "totalPrice": 40.00  
10  },  
11   {  
12     "no": 2,  
13     "productId": "FG0A-CLEAR-OPPOA3-B",  
14     "materialId": "FG0A-CLEAR",  
15     "modelId": "OPPOA3-B",  
16     "qty": 1,  
17     "unitPrice": 40.00,  
18     "totalPrice": 40.00  
19   },  
20   {  
21     "no": 3,  
22     "productId": "WIPING-CLOTH",  
23     "qty": 2,  
24     "unitPrice": 0.00,  
25     "totalPrice": 0.00  
26   },  
27   {  
28     "no": 4,  
29     "productId": "CLEAR-CLEANER",  
30     "qty": 2,
```

```
31     "unitPrice": 0.00,  
32     "totalPrice": 0.00  
33   }  
34 ]
```

---

### Case 5: One bundle product with wrong prefix and split by / symbol into three product

#### Input

```
1  [  
2    {  
3      "no": 1,  
4      "platformProductId": "FG0A-CLEAR-OPPOA3/%20xFG0A-CLEAR-OPPOA3-B/FG0A-MAT  
5      "qty": 1,  
6      "unitPrice": 120,  
7      "totalPrice": 120  
8    }  
9  ]
```

## Expected Output

```
1  [
2    {
3      "no": 1,
4      "productId": "FG0A-CLEAR-OPPOA3",
5      "materialId": "FG0A-CLEAR",
6      "modelId": "OPPOA3",
7      "qty": 1,
8      "unitPrice": 40.00,
9      "totalPrice": 40.00
10   },
11   {
12     "no": 2,
13     "productId": "FG0A-CLEAR-OPPOA3-B",
14     "materialId": "FG0A-CLEAR",
15     "modelId": "OPPOA3-B",
16     "qty": 1,
17     "unitPrice": 40.00,
18     "totalPrice": 40.00
19   },
20   {
21     "no": 3,
22     "productId": "FG0A-MATTE-OPPOA3",
23     "materialId": "FG0A-MATTE",
24     "modelId": "OPPOA3",
25     "qty": 1,
26     "unitPrice": 40.00,
27     "totalPrice": 40.00
28   },
29   {
30     "no": 4,
31     "productId": "WIPING-CLOTH",
32     "qty": 3,
33     "unitPrice": 0.00,
34     "totalPrice": 0.00
35   },
36   {
37     "no": 5,
38     "productId": "CLEAR-CLEANNER",
39     "qty": 2,
40     "unitPrice": 0.00,
41     "totalPrice": 0.00
42   },
43   {
44     "no": 6,
45     "productId": "MATTE-CLEANNER",
46     "qty": 1,
47     "unitPrice": 0.00,
```



```
48     "totalPrice": 0.00
49   },
50 ]
```

---

### Case 6: One bundle product with wrong prefix and have / symbol and \* symbol

#### Input

```
1  [
2    {
3      "no": 1,
4      "platformProductId": "--FG0A-CLEAR-OPPOA3*2/FG0A-MATTE-OPPOA3",
5      "qty": 1,
6      "unitPrice": 120,
7      "totalPrice": 120
8    }
9  ]
```

## Expected Output

```
1  [
2    {
3      "no": 1,
4      "productId": "FG0A-CLEAR-OPPOA3",
5      "materialId": "FG0A-CLEAR",
6      "modelId": "OPPOA3",
7      "qty": 2,
8      "unitPrice": 40.00,
9      "totalPrice": 80.00
10   },
11   {
12     "no": 2,
13     "productId": "FG0A-MATTE-OPPOA3",
14     "materialId": "FG0A-MATTE",
15     "modelId": "OPPOA3",
16     "qty": 1,
17     "unitPrice": 40.00,
18     "totalPrice": 40.00
19   },
20   {
21     "no": 3,
22     "productId": "WIPING-CLOTH",
23     "qty": 3,
24     "unitPrice": 0.00,
25     "totalPrice": 0.00
26   },
27   {
28     "no": 4,
29     "productId": "CLEAR-CLEANER",
30     "qty": 2,
31     "unitPrice": 0.00,
32     "totalPrice": 0.00
33   },
34   {
35     "no": 5,
36     "productId": "MATTE-CLEANER",
37     "qty": 1,
38     "unitPrice": 0.00,
39     "totalPrice": 0.00
40   }
41  ]
```

## Case 7: one product and one bundle product with wrong prefix and have / symbol and \* symbol

### Input

```
1  [
2    {
3      "no": 1,
4      "platformProductId": "--FG0A-CLEAR-OPPOA3*2/FG0A-MATTE-OPPOA3*2",
5      "qty": 1,
6      "unitPrice": 160,
7      "totalPrice": 160
8    },
9    {
10     "no": 2,
11     "platformProductId": "FG0A-PRIVACY-IPHONE16PROMAX",
12     "qty": 1,
13     "unitPrice": 50,
14     "totalPrice": 50
15   }
16 ]
```

### Expected Output

```
1  [
2    {
3      "no": 1,
4      "productId": "FG0A-CLEAR-OPPOA3",
5      "materialId": "FG0A-CLEAR",
6      "modelId": "OPPOA3",
7      "qty": 2,
8      "unitPrice": 40.00,
9      "totalPrice": 80.00
10   },
11   {
12     "no": 2,
13     "productId": "FG0A-MATTE-OPPOA3",
14     "materialId": "FG0A-MATTE",
15     "modelId": "OPPOA3",
16     "qty": 2,
17     "unitPrice": 40.00,
18     "totalPrice": 80.00
19   },
20   {
21     "no": 3,
22     "productId": "FG0A-PRIVACY-IPHONE16PROMAX",
23     "materialId": "FG0A-PRIVACY",
```

```
24     "modelId": "IPHONE16PROMAX",
25     "qty": 1,
26     "unitPrice": 50.00,
27     "totalPrice": 50.00
28 },
29 {
30     "no": 4,
31     "productId": "WIPING-CLOTH",
32     "qty": 5,
33     "unitPrice": 0.00,
34     "totalPrice": 0.00
35 },
36 {
37     "no": 5,
38     "productId": "CLEAR-CLEANNER",
39     "qty": 2,
40     "unitPrice": 0.00,
41     "totalPrice": 0.00
42 },
43 {
44     "no": 6,
45     "productId": "MATTE-CLEANNER",
46     "qty": 2,
47     "unitPrice": 0.00,
48     "totalPrice": 0.00
49 },
50 {
51     "no": 7,
52     "productId": "PRIVACY-CLEANNER",
53     "qty": 1,
54     "unitPrice": 0.00,
55     "totalPrice": 0.00
56 }
57 ]
```