

Задание:

Клиент хочет запустить магазин по продаже пива. Бутылки пива бывают разных размеров. Упаковка(коробка) бывает тоже разных размеров и может включать в себя: 6,8,10,12,16,25 ячеек.

Пиво может занимать: 0.5 или 1 ячейку.

Необходимо написать функцию, которая будет разбивать входящий массив пива по ящикам.

Пример:

Заказ занимает 15 ячеек → функция возвращает 1 коробку на 16 ячеек

Заказ занимает 26 ячеек → 16 + 10 .

Важно понимать, что оптимальное решение должно находить вариант с максимальной заполненностью коробок. То есть 26 → 25 + 1 , это неприемлемый вариант так как одна из коробок будет иметь 5 пустых ячеек.

На входе дается List<Map> содержащий код пива, его “емкость” и кол-во. На выходе ожидается List<Map> содержащий размер коробки, пиво внутри.

Код приведен только как пример для общего понимания того, что ожидается. Реализация не ограничивается строгим использованием данных классов и объектов. Допускается использование кастомных классов.

```
List<Map> divideOrder(List<Map> beer) {  
    ....  
}
```

Example:

input:

```
[  
    [sku: "beer1", cell: 1, quantity: 3], [sku: "beer2", cell: 0.5, quantity: 2], [sku: "beer3", cell:  
    1, quantity: 1]  
]
```

result:

```
[  
    [boxPack: 6,  
    content: [[name: "beer1", quantity: 3], [name: "beer2", quantity: 2], [name: "beer3",  
    quantity: 1]]  
    ]  
]
```

input:

```
[  
    [sku: "beer1", cell: 1, quantity: 15], [sku: "beer2", cell: 0.5, quantity: 3], [sku: "beer3", cell:  
    1, quantity: 10]  
]
```

result:

```
[  
    [boxPack: 16,  
    content: [[name: "beer1", quantity: 15],[name: "beer3", quantity: 1]]  
    ],  
    [boxPack: 12,
```

```
],  
  content: [[name: "beer3", quantity: 9],[name: "beer2", quantity: 3]]  
]
```