Software Engineering Challenge

Imagine for a moment that one of our product lines ships in various pack sizes:

- 250 Items
- 500 Items
- 1000 Items
- 2000 Items
- 5000 Items

Our customers can order any number of these items through our website, but they will always only be given complete packs.

1. Only whole packs can be sent. Packs cannot be broken open.
2. Within the constraints of Rule 1 above, send out no more items than necessary to fulfil the order.
3. Within the constraints of Rules 1 & 2 above, send out as few packs as possible to fulfil each order.

(Please note, rule #2 takes precedence over rule #3)

So, for example:

Items ordered	Correct number of packs	Incorrect number of packs
1	1 x 250	1 x 500 – more items than necessary
250	1 x 250	1 x 500 – more items than necessary
251	1 x 500	2 x 250 – more packs than necessary
501	1 x 500 1 x 250	1 x 1000 – more items than necessary 3 x 250 – more packs than necessary
12001	2 x 5000 1 x 2000 1 x 250	3 x 5000 – more items than necessary

To further illustrate the rules above, please consider this custom pack size example:

- 23 Items
- 31 Items
- 53 Items

Items Order: 263
Correct Number of packs: 2x23, 7x31
Incorrect answer: 5x53

Write an application that can calculate the number of packs we need to ship to the customer.

The API must be written in Golang & be usable by a HTTP API (by whichever method you choose) and show any relevant unit tests.

Optional:

- Keep your application flexible so that pack sizes can be changed and added and removed without having to change the code.
- Create a UI to interact with your API

Please also send us your code via a publicly accessible git repository, GitHub or similar is fine, and deploy your application to an online environment so that we can access it and test your application out.

We look forward to receiving your application!