A Problem Squared Zero Three Six = Score Menu Sets

Arbitrarily large orders that will make your waiter sweat in computation.

Maths (not rigorous, a physicist's job)

Menu of size N (i.e. N different items are in the menu) let's have $S = s_0, ..., s_i, ..., s_{N-1}$ the set all menu items.

We note α_i the desired quantity of the ith item.

To write it with a specific base, we simply need to split it. The number of times the number must be repeated with quantity $(k-1)$:

$u_i = \alpha_i // (k-1)$ (with $//$ being integer division).

The position of the menu item being i, to add a second instance of it, we'll place it naturally at $S + i$, then at $S * 2 + i$ etc... u_i times in total (counting $S * 0$).
Then, another repeat of quantity $\alpha_i \mod (base - 1)$ with exponent $S * u_i + i$.

The ith item number is:

$O_i = \sum_{r=0}^{u_i - 1} (k-1) k^(r * S + i) + (\alpha_i \mod (base - 1))$

then, your order is a sum of all item-numbers : $O = \sum_{i=0}^{n} O_i$

I hope Dexter can confirm my working out is in order.

Python (a random person offering you to run their code on your computer? Not suspicious at all)

simply run menu.py, I did not bother creating a nice interface, sorry. Only two inputs: the base and the menu order array that looks like this:

```
my_order=[0,0,2,15,0,3,0]
```

The output should be something like:

```
> python menu.py
Order with classical quantities: [0, 0, 2, 15, 0, 3, 0]
Base chosen: 5
Order with computed quantities and item index for each item:
[0, 0, 50, 178816986123047375, 0, 9375, 0]
Final order: 178816986123056800

### I changed the source to use the sanity check

> python menu.py
Order with classical quantities: [1, 1, 1, 1, 1, 1, 1]
Base chosen: 5
Order with computed quantities and item index for each item:
[1, 5, 25, 125, 625, 3125, 15625]
Final order: 19531
```