

# AI/ML Engineering Mini Project



## Problem Statement

Consider the following situation: A supplement company has only one product, but it offers them in a variety of unique packaging sizes indicated by the number of supplied servings (for example, 10 days, 14 days, 30 days). Depending on the required servings for each customer, the company combines a variety of its packages to fulfill the customer's order.

Due to the limited number of packaging sizes available, often the company has to ship more than the original order volume to fulfill the customer's request. An order is called *perfectly purchasable* if it can be exactly fulfilled with a combination of multiple packages. For example, assuming the company only offers two different packages for 5 days and 14 days support, an order request for 19 days is *perfectly purchasable*, but an order request for 21 days is not. Your task is to write a program that can find the largest order volume that is **NOT** *perfectly purchasable*.

In other words, given a list of positive integer numbers, write a program that returns the largest number that cannot be expressed as a linear combination of input values with non-negative integer weights.

You can work with the programming language of your choice, but using external libraries/modules are not allowed. Please include an executive summary of the overall approach as well as test cases considered to solve the problem in your submission. Your report should include your thought process on how the solution/algorithm is

designed and why it works in different cases. Please include your reasoning to prove the convergence of the algorithm in all cases.