Advanced Algorithms
Exercises 2
DI – NOVA FCT

1. Recall that, if S is a nonempty set of integers, the *sum* of S is:

$$\text{sum}(S) = \sum_{x \in S} x \,.$$

The *sum* of the empty set is zero: $\text{sum}(\emptyset) = 0$.

In the **Maximum Subset-Sum Problem**, we are given a set P of positive integers and a positive integer M. The goal is to find a subset of P whose sum is as large as possible but not larger than M. Without loss of generality, let us assume that no element of P is greater than M.

For example, if $P = \{15, 10, 14, 19\}$ and $M = 35$, the optimal solution is $\{15, 19\}$ and its sum is 34.

(a) Consider algorithm **greedySS1**.

```
greedySS1( P = {x_1, x_2, ..., x_n}, M )
    X = ∅
    s = 0
    for i = 1, ..., n
        if s + x_i ≤ M
            X = X ∪ {x_i}
            s = s + x_i
    return X
```

Is **greedySS1** a ρ-approximation algorithm for the Maximum Subset-Sum Problem for some constant ρ? Justify your answer.

(b) Now consider algorithm **greedySS2**.

```
greedySS2( P = {x_1, x_2, ..., x_n}, M )
    X = ∅
    s = 0
    for i = 1, ..., n
        if s + x_i ≤ M
            X = X ∪ {x_i}
            s = s + x_i
        else
            if s ≥ x_i
                return X
            else
                return {x_i}
    return X
```

Prove that **greedySS2** is a 2-approximation algorithm for the Maximum Subset-Sum Problem.

2. [Kleinberg and Tardos 2005] Suppose you are acting as a consultant for a Port Authority. When a ship arrives, with n containers of weights $w_1, w_2, \ldots, w_n$, there are trucks standing on the dock and each of them can hold C units of weight. Assume that $w_1, w_2, \ldots, w_n$ and C are positive integers (and $w_i \leq C$, for every $i = 1, 2, \ldots, n$). You can stack several containers in a truck, provided that the weight capacity C is not exceeded. The goal is to minimize the number of trucks used to carry all the containers.

   You know that the corresponding decision problem is NP-complete. So, you are considering using the following greedy algorithm:

   - Start with an empty truck and begin piling containers $1, 2, \ldots$ into it until you reach a container k that would overflow the truck capacity. At that moment, declare the truck loaded and send it off.
   - Then, continue the process (from container k) with a fresh truck.

   This algorithm, by considering a truck at a time, may not minimize the number of trucks used to carry all containers.

   (a) Give an instance where this algorithm does not use the minimum possible number of trucks.

   (b) Show, however, that this algorithm is a 2-approximation algorithm for the problem.
   Hint: Consider the load carried by consecutive (disjoint) pairs of trucks in the computed solution. Then, analyse separately the cases in which the number of trucks used by the algorithm is even and odd: