- Always try to give algorithm with best possible running time. The points that you obtain will depend on the running time of your algorithm. For example, a student who gives an $O(n)$ algorithm will receive more points than a student who gives an $O(n^2)$ algorithm.
- You are required to give proofs of correctness whenever needed. For example, if you give a greedy algorithm for some problem, then you should also give a proof why this algorithm outputs optimal solution.
- **Use of unfair means will be severely penalized.**

There are 3 questions for a total of 40 points.

---

(10) 1. Given integers $R_1, R_2, ..., R_n$ and $C_1, C_2, ..., C_n$, design an algorithm that determines if there exists an $n \times n$ 0/1 matrix A such that:

   1. For all i, $\sum_{j=1}^{n} A[i,j] = R_i$, and
   2. For all j, $\sum_{i=1}^{n} A[i,j] = C_j$.

(15) 2. Recall the Fractional Knapsack problem discussed in the class. Suppose we remove the "fractional" assumption from the problem. That is, for every item, you can either take the item or not take it (recall in the fractional version, you were allowed to any any arbitrary fraction of the item). The resulting problem turns out to be a hard problem (we will discuss this when we talk about computational intractability). So, makes sense to design approximation algorithms for this problem. Consider the following greedy algorithm for the problem.
   GreedySteal

   - Let $\{i_1, ..., i_n\}$ be the items sorted in decreasing order of value per unit volume (i.e., $V(i)/W(i)$).
   - Let T be the item with maximum value.
   - Let k be such that $\sum_{i=1}^{k} W(i) \leq W$ and $\sum_{i=1}^{k+1} W(i) > W$.
   - Either steal items $\{i_1, ..., i_k\}$ or item T whichever maximizes the total value.

   Let G be the value of the items chosen by the greedy algorithm and let OPT be the value of the optimal solution for the above problem. Show that $G \geq (1/2) \cdot OPT$.

(15) 3. You are given n points on a two-dimensional plane. A point (x, y) is said to *dominate* a point (x', y') iff $x > x'$ and $y > y'$. Design an algorithm that outputs all points that are not dominated by any other point.