There are 1 questions for a total of 15 points.

---

(15) 1. Graph feasibility: Given positive integers $(d_{out}[1], d_{out}[2], ..., d_{out}[n])$ and $(d_{in}[1], d_{in}[2], ..., d_{in}[n])$, implement the greedy algorithm you designed for problem 6 in HW-3 that determines whether there exists a simple directed graph where the ith node has indegree $d_{in}[i]$ and outdegree $d_{out}[i]$. (*Recall that a simple directed graph is a graph that does not have self-loops or multi-edges.*)

Your algorithm should also output a valid adjacency matrix for the graph in case it exists. This means that the output matrix should be such that the sum of ith row should be $d_{out}[i]$, and the sum of the ith column is $d_{in}[i]$. Your program should take input from a file named **input.txt** and should write the output in a file named **output.txt**. Your program should produce an output within 15 seconds for this assignment. The format for input and output files is as follows.

INPUT: The first line of the input file gives the value of n (assume $n \leq 1000$). The next line gives the value of $d_{out}[1], d_{out}[2], ..., d_{out}[n]$ separated by commas. The final line gives the value of $d_{in}[1], d_{in}[2], ..., d_{in}[n]$ separated by commas. Below is an example of an input file.

```
3
1,1,1
1,1,1
```

OUTPUT: The first line of the output should indicate whether such a matrix A exists. This is a 0/1 value. In case, this line contains 0, then the subsequent lines should be empty (since your algorithm thinks that no such matrix can exist). In case the first line is 1, then the subsequent lines should give the matrix. The second line should give the first row of the matrix (entries separated by commas), the third line should give the second row of the matrix and so on. For example, consider the output file corresponding to the input file above:

```
1
0,1,0
0,0,1
1,0,0
```

Note: There are **no** whitespaces or newline characters at the end of the output file. Having such whitespaces might lead to the autograder marking the output as incorrect. There may be multiple possible solutions in case a matrix exists for a given input. In such cases, we will verify your answer by counting the number of 1's in rows and columns and cross-checking with the input.

SUBMISSION INSTRUCTIONS: All your program files should be in a directory **hw3_prog**. You will be asked to create a zip of this directory and submit this zip file. In this directory, there should be a makefile that will compile your code (read about makefile on the net in case you do not know what it is). After running **make**, the directory should have an executable called **checkmatrix**. This, when executed, should read the input file (**input.txt**) and write the answer in the output file (**output.txt**).