Instructions:

We will model a facility assignment problem in this lab, similar to the assignment problem discussed in the previous lab.
Recall that two-dimensional decision variables are useful in the assignment problems, and we will focus on using such two-dimensional variables in model building using `pyomo`.
We shall also check the solve time required to solve mixed integer linear programs when compared to solving linear programs.

We will continue to model problems by loading information from files in this lab. We will also continue modeling problems with integer variables, wherever required.

Recall, to load directly from a file with comma separated values (`.csv` file), we used `pandas` library. The construct `pandas.read_csv` helps to read contents from a `.csv` file. Please check https://pandas.pydata.org/pandas-docs/stable/getting_started/index.html to know more about `pandas` library.

In this lab too, `iloc` can be used to access data from pandas data frame. For more details on this `iloc` function, please see https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.iloc.html

Please follow the instructions given below:

- Please use different notebooks for solving different problems.
- The notebook name for Exercise 1 should be `YOURROLLNUMBER_IE507_Lab7_Ex1.ipynb`.
- Similarly, the notebook name for Exercise 2 should be `YOURROLLNUMBER_IE507_Lab7_Ex2.ipynb`.
- Please discuss your doubts with TAs so that they can clarify.

For more details on pyomo, please consult https://pyomo.readthedocs.io/en/stable/index.html.

There are only 2 exercises in this lab. Try to solve all problems on your own. If you have difficulties, ask the TAs or Instructor.

Only the questions marked **[R]** need to be answered in the notebook. You can either print the answers using `print` command in your code or you can write the text in a separate text tab. To add text in your notebook, click `+Text`. Some questions require you to provide proper explanations; for such questions, write proper explanations in a text tab.

After completing this lab's exercises, click File → Download .ipynb and save your files to your local laptop/desktop. Create a folder with name `YOURROLLNUMBER_IE507_Lab7` and copy your `.ipynb` files to the folder. Also copy the `.csv` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE507_Lab7.zip`. Then upload only the `.zip` file to Moodle.

The deadline for today's lab submission is **5 PM today**.

Exercise 1: Assigning Facilities to Locations [20 Marks]
In the previous lab, we discussed the problem of assigning different factories to different locations. Let us now consider a different variation of the assignment problem discussed in the previous lab. Once the 12 factories are set up and start operating, they will be transporting goods amongst themselves. We would like to assign locations to these factories so that the total cost of transportation between them is minimized. The quantities (in Tonnes) that must be transported every week from factory-i to factory-j, $i, j = 1, \ldots, 12$ are given in the Table 1 below. The cost of transporting goods depends on the location of the source and destination factory. Table 1 below gives the cost of transporting one tonne of goods from location i to location j, $i, j = 1, \ldots, 12$. We can ignore the setup costs that were used in the previous lab.

Factory (source)	Factory (Destination)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	18	22	19	23	20	18	19	0	20	24	35
2	21	0	20	0	19	0	22	20	18	19	0	33
3	22	23	0	0	0	17	16	24	16	18	24	0
4	17	25	0	0	21	22	20	17	15	16	24	31
5	12	19	18	19	0	21	21	23	21	0	0	21
6	20	0	0	17	21	0	20	0	19	17	22	20
7	22	24	28	16	23	0	0	19	24	25	30	19
8	23	29	20	17	24	24	23	0	19	22	30	26
9	0	28	29	21	0	24	18	18	0	0	22	23
10	31	20	0	19	23	0	20	24	19	0	20	0
11	17	0	25	23	20	18	16	19	0	39	0	34
12	0	32	24	26	0	19	0	18	37	21	20	0

Table 1: Quantities that must be transported from a factory to another factory every week

Location (Source)	Location (Destination)											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	11	13	9	19	20	27	13	19	11	19	10
2	11	0	8	9	11	19	10	15	12	23	24	11
3	13	8	0	18	19	19	27	27	19	24	12	17
4	9	9	18	0	19	20	10	20	21	20	27	10
5	19	11	25	19	0	21	17	26	20	14	24	17
6	20	13	20	20	21	0	28	14	22	17	22	23
7	27	10	17	10	17	28	0	12	18	26	10	12
8	13	15	27	20	19	14	12	0	27	10	19	17
9	29	12	19	21	20	22	18	27	0	20	22	16
10	11	20	24	10	14	17	26	10	10	0	25	21
11	18	19	12	14	14	22	10	19	12	25	0	21
12	20	21	17	10	17	20	22	17	16	21	21	0

Table 2: Unit costs of transportation from a location to another location

1. **[R]** Write a mathematical model to solve the assignment problem explained above. Define all the variables and constraints clearly. Use appropriate notations and define appropriate sets to be used in your optimization problem.
2. **[R]** If there are n factories and n locations, how many variables and constraints are there in your model?

3. Construct a `pyomo` model for this problem for a general n. You can assume that the quantitites matrix and unit cost matrix are given as data from `txt` files and can be loaded as `numpy` arrays.
4. Use the data in Table 1 and Table 2 to make `.txt` files for your model. You can use the `quantities.txt` file and `unitcosts.txt` file uploaded in Moodle to create the `.txt` files. Name the files as `lab7_ex1_q.txt` and `lab7_ex1_c.txt`.
5. Copy the files to colab environment.
6. Use `numpy.loadtxt` to load the data from `lab7_ex1_q.txt` and `lab7_ex1_c.txt` files into `numpy` arrays.
7. Adapt the general `pyomo` model you created, to use the data loaded from the `lab6_ex2_q.txt` and `lab7_ex1_c.txt` files.
8. Use `cbc` solver to solve your optimization problem. In your code, remember to specify which variables are integers.
9. [**R**] Solve the problem and report which facility must be opened at each location.
10. [**R**] Now change the integer variables in your model to continuous variables, and re-solve the problem. Report the solution (only the non-zero values of the solution).
11. [**R**] Are the optimal costs for both problems same? Are the values of the variables still integer-valued? If yes, explain why.
12. **IMPORTANT:** Do not forget to upload the data files with your submission.

Exercise 2: Time taken in MILP. [**10 marks**] In this exercise we will compare the time taken to solve LP and MILP. Consider the integer optimization problem

$$\begin{aligned} \max \; & \sum_{j=1}^{n} c_j x_j \\ \texttt{s.t.} \; & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \qquad i = 1, \ldots, m \\ & x_j \in \{0, 1\}, \quad j = 1 \ldots, n. \end{aligned} \tag{MILP}$$

and its LP relaxation

$$\begin{aligned} \max \; & \sum_{j=1}^{n} c_j x_j \\ \texttt{s.t.} \; & \sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1, \ldots, m \\ & 0 \le x_j \le 1, \quad j = 1 \ldots, n. \end{aligned} \tag{LP}$$

Specific data is available for this problem in Moodle in `lab7_ex2_lp_ip_coef.txt` file. It has $n = 500$ variables and $m = 5$ constraints.

1. Copy the `lab7_ex2_lp_ip_coef.txt` file to colab environment.
2. Use `numpy.loadtxt` to load `lab7_ex2_lp_ip_coef.txt` into an appropriate `numpy` array.
3. [**R**] Create a pyomo model using the coefficients loaded from the file and solve the MILP using `cbc` solver. Report the optimal objective function value (Do not report the values of variables).
4. [**R**] How much time does it take to solve the MILP? Suppose you have saved the results of `opt_cbc.solve` to `results`, you can use `results.solver.time` to report time.
5. [**R**] Solve the LP relaxation (LP obtained by removing integer restrictions on all variables) of the same problem using `cbc` solver. Report only the objective function values of the optimal solution. (You need not report the values of the variables.)
6. [**R**] How much time does it take to solve the problem? Compare with the results obtained by solving the MILP.
7. [**R**] Also report the statistics printed in Branch and Bound section of `results` for both MILP and LP.