Instructions:
In this session, we will continue to use Pyomo modeling language and software. We will see more useful constructs which will help in modelling problems which have many variables and constraints. Please follow the instructions given below:

- Please use different notebooks for solving different problems.
- The name of your notebook for Exercise 1 should be `YOURROLLNUMBER_IE507_Lab2_Ex1.ipynb`. Similarly, the name of your notebook for Exercise 2 should be `YOURROLLNUMBER_IE507_Lab2_Ex2.ipynb`, etc.
- Please post your doubts in MS Teams or Moodle so that TAs can clarify.

For more details on pyomo, please consult https://pyomo.readthedocs.io/en/stable/index.html.
For details on `numpy` python package, please consult https://numpy.org/doc/stable/numpy-ref.pdf. This file has been uploaded in moodle as well.
There are only 3 exercises. Try to solve all problems on your own. If you have difficulties, ask the Instructors or TAs.
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`.

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_Lab2` and copy your `.ipynb` files to the folder. Then zip the folder to create `YOURROLLNUMBER_IE507_Lab2.zip`. Then upload only the `.zip` file to Moodle. Recall that these instructions were also provided in a video posted for Lab 1.
The deadline for today's lab submission is **tomorrow, 11.59 PM IST**.

Exercise 1: [6 Marks]
In the practice problem, we saw that python `numpy` arrays can be used to store the coefficients of the objective function and constraints. In this exercise, we will consider a slight variation and consider the problem given below.

$$\begin{aligned}
\min\ c_1x_1 + c_2x_2 + \ldots + c_Nx_N&, \\
x_1 + x_2 + \ldots + x_N &\leq b_1, \\
w_1x_1 + \ldots + w_Nx_N &= b_2, \\
x_i &\leq u_i, \quad i = 1, \ldots, N, \\
x_i &\geq l_i, \quad i = 1, \ldots, N.
\end{aligned}$$

1. Now suppose we want to solve the problem with 12 variables. Modify the code in practice problem so that you can solve the problem given above with 12 variables. You may assume that

$$\begin{aligned}
&b_1 = 161.2, \\
&b_2 = 54.6, \\
&c = [-8.1, 10.15, 30.5, 50.05, 0.05, 80.5, -0.25, -31.02, 50.65, 0.725, -0.8, 100.6]; \\
&w = [0, 1.6, -2.02, 0, 1.01, -2.005, 0, 1.39, 1, -1.214, 0, -5.32]; \\
&l = [-\infty, 1, 0, 0, 7, 0, -\infty, -\infty, 1, 1, -5, 2], \\
&u = [4, 3, +\infty, 2, 10, +\infty, 13, 20, +\infty, 21, 5, 60]
\end{aligned}$$

2. **[R]** Solve the new problem and report the optimal solution and the value of the objective function (or cost).

Exercise 2: [8 Marks] We will now consider another general formulation of the form given below:

$$\begin{aligned} \min\ & c^\top x, \\ \text{s.t } & Ax \leq b \\ & l \leq x \leq u. \end{aligned}$$

where c, x, l and u are $N \times 1$ vectors and $c^\top x = \sum_{i=1}^{N} c_i x_i$. A is a $M \times N$ matrix and b is $M \times 1$ vector. $Ax \leq b$ results in M constraints and note that i-th row of $Ax \leq b$ can be written as a constraint of the form:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{iN}x_N \leq b_i.$$

We will solve the general form using concrete values given below:

$$\begin{aligned} c &= [-9.6, -2.85, 41.5, 33.05, -0.75, 20.5, -0.35, -31.02, 50.65, 6.125]; \\ A &= \begin{pmatrix} 0 & 1 & -2 & 0 & 1 & -2 & 0 & 1 & 1 & -1 \\ -1 & 1 & -3 & 2.5 & 0 & 1.4 & 0 & -2 & 1 & -1.5 \\ -0.45 & 4.78 & -2 & -3.22 & -1.3 & 1 & -1 & 0 & 0 & 0.25 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 2 & 3 & 0 \end{pmatrix} \\ b &= [35.4, 47.9, 41.43, 22.7] \\ l &= [-\infty, -\infty, 0, 0, -67, 0, -23, -5, 6, 1], \\ u &= [4, \infty, +\infty, 2, 20, 55, 23, 15, +\infty, +\infty] \end{aligned}$$

We will now see some instructions to prepare your code to solve this general formulation.
Preparing the code for solving the general formulation:

- Recall that we created a variable `N` to denote the number of variables. Similarly create a variable `M` to denote the number of constraints. Use appropriate value for `M` based on the concrete problem given above.
- Recall that we used two arrays named `constr1_coef_p` and `constr2_coef_q` for storing the coefficients of the two constraints. Now, instead we will create a single two-dimensional `numpy` array with name `constr_coef_A` to store the coefficients of all constraints. Check the `numpy` documentation and see how to create a two-dimensional `numpy` array. Create the array with the values given in the matrix A given above.
- Similarly we used two variables `constr1_rhs_b1` and `constr1_rhs_b2` for storing the values used in RHS of first and second constraints respectively. Now, instead we will create a single `numpy` array `constr_rhs_b` and store the values in b array given above.
- Also recall that we used a `ConstraintList()` for adding the constraints. We added each constraint separately. Instead of adding each constraint separately, we will try to add the constraints in a loop. For this purpose, let us create a list using the following command

  ```
  row_indices = np.arange(M)
  ```

  Check the contents of this list using a `print` statement. Now create a `for` loop to load the constraint to the list `ConstraintList()`. Your code should be of the form as given below:

  ```
  for <fill in appropriate syntax>:
      model.constraints.add(<fill in appropriate syntax>)
  ```

- Modify the upper bound and lower bound arrays in your code appropriately with the values given above in l and u arrays.
- Print the model to check if the constraints have been added properly.
- You are now ready to answer the questions.

Question:

- [**R**] Solve the problem and report the optimal solution and the value of the objective function (or cost).
- [**R**] Which constraints are active and which constraints are inactive? Explain with proper reasons.

Exercise 3: Alloy Mix Problem. [15 Marks]
Leslie Metal Co. produces 450 tons of metallic alloy product of composition: 35% Chromium, 45% Copper, and 20% Silver. In order to produce this alloy, it can mix 10 available alloys: A-1, A-2, ..., A-10. The alloys have the following composition and costs.

	A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8	A-9	A-10
% of Chromium	15	15	10	20	25	10	50	15	30	55
% of Copper	80	75	75	60	55	55	40	35	30	30
% of Silver	5	10	15	20	20	35	10	50	40	15
in-house stock (Tons)	12	9	16	25	4	5	10	13	0	6
in-house cost (per Ton)	35	50	58	60	44	39	45	55	35	40
purchase cost (per Ton)	72	95	110	125	88	74	95	115	60	84

The company can either use the stocks of raw material alloys available in the company or purchase them from market. The stocks available in the company are limited while the quantity available in market is unlimited. You are given the task of finding how much of each raw-material should be put in the final mix to obtain 450 Tons of the product.

1. **[R]** Formulate an LP model that can be solved to find the optimal mix of raw-material alloys.
2. Modify the code used in Exercise 2 with appropriate number of variables, number of constraints, coefficient arrays, lower bound and upper bound arrays to construct a model for this problem.
3. **[R]** Solve the model and report the optimal value and the optimal solution.
4. **[R]** Among the alloys that need to be bought from the market, which three alloys are ranked high in terms of the quantities to be bought and which three alloys are ranked low?
5. **[R]** Is there any alloy whose stock is completely utilized? Why?
6. **[R]** Is there any alloy which need not be acquired from market? Why?