Solving Linear Programming Problems in Python

(The Simplex Algorithm)

1. Setup the Environment

This assumes that you already have python installed.
The Program was tested using Python 3.9.1 installed using Anaconda Distrubution.

Though it will work with any version of python 3.

a. If you are using Anaconda Distrubution

- Create the environment and install the requirements

```
conda create --name lpp_env --file requirements.txt
```

- Activate the Enviroment

```
conda activate lpp_env
```

b. If you like python virtual environment (Not Tested)

Basically, please make sure that all the dependencies are installed before running the main.py file.

- Create a Virtualenv

```
python3 -m venv env
```

- Activate the env

```
.\env\Scripts\activate
```

- Install the Dependencies

```
pip install -r requirements.txt
```

or you can directly install numpy

```
pip install numpy
```

2. To run the program run the command in the terminal

```
python main.py -i input_filename
```

eg:

```
python main.py -i input.txt
```

When a problem is finished, the program will pause. Press any key to continue on to the next problem.

Changing the inputs to the program

All the input problems are stored in a text file which we pass to the main.py when we run the program.

Rules for the input text file

- A line beginning with a # is treated as a comment and will be ignored.

- Leave a single blank line between two LPPs.

- The variables can be anything. eg: x_1 and x_2, x1 and x2, a and b, or x and y, etc.

- Please leave no space between the coefficient and the variable. Anything after a number will be treated as a variable.
  eg:
  52x_1: coefficient 52 , variable x_1
  52x1: coefficient 52 , variable x1

- Please use the signs >=, <= or = for the constraints.
  Do not use ≥ or ≤.

- Example Format of LPP:

  ```
      Maximize: z = 5x_1 + 4x_2
      Subject to:
          6x_1 + 4x_2 <= 24
          x_1 + 2x_2 <= 6
          -x_1 + x_2 <= 1
          x_2 <= 2
  ```

Note 1: The program is designed to solve "Easy LPPs" (i.e., LPPs which are a format of A X ≤ b and X ≥ 0, b ≥ 0).
The program also assumes non-negativity constraints.

Note 2: The Program uses unicode characters in some places like "||" symbol is used for table borders and "α" is used when the LPP has Alternative Optimums. If the terminal cannot handle unicode, it'll display � or □ or something similar of sorts.

Note 3: The table folder contains code used to print the Simplex Tableau. The code is a part of [DashTable] Library and is available at (https://github.com/doakey3/DashTable). Only the reuqired files are kept and the original code has been modified a bit.

Modifications Done:

- **Use Different character for the Table Border**
- **Mark First two rows as Table Header**
- **Add Padding to individual cells**