# PROJECT REPORT

Erkam Kavak-2020400174

Emre Batuhan Göç-2020400090

Muhammet Tayyip Kamiloğlu-2019400111

Hüseyin Çivi-2019400165

# Introduction

This project's aim was to implement a Simplex algorithm without using any linear algebra libraries using python. Some example data files containing input constraints were given to the algorithm and an optimal solution was printed out.

# Terms

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{s.t.} \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i = 1, \dots, m$$

$$x_j \geq 0, \qquad j = 1, \dots, n$$

$$\begin{aligned} &\min c^T x \\ &\text{s.t.} \quad Ax \leq b \\ &\qquad x \geq 0 \end{aligned}$$

A, is the matrix containing the coefficients of the constraints.

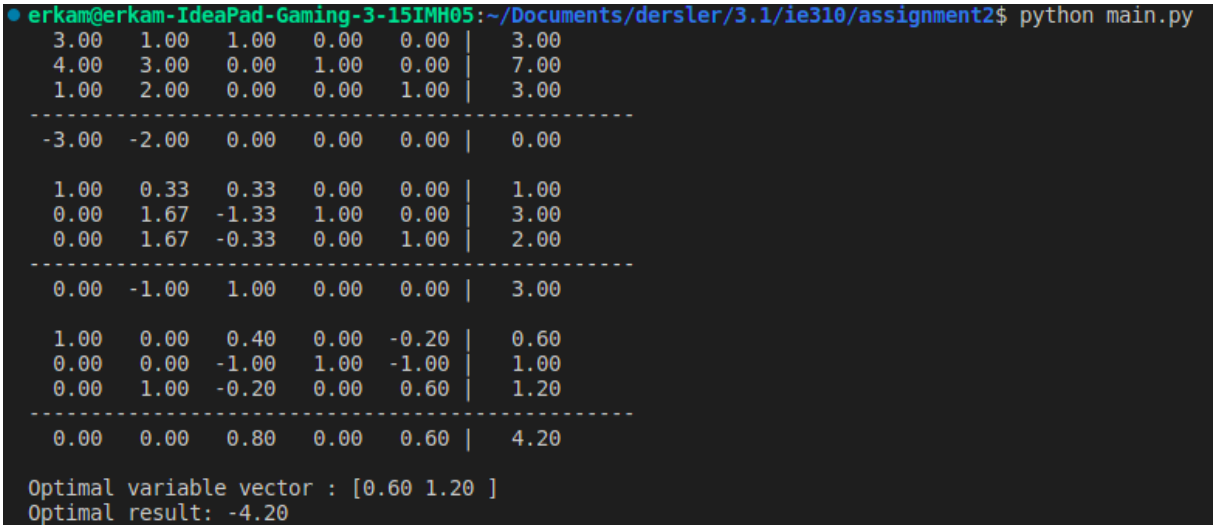
b, is the vector consisting of the right-hand side variables.

c, is the vector consisting of objective function coefficients.

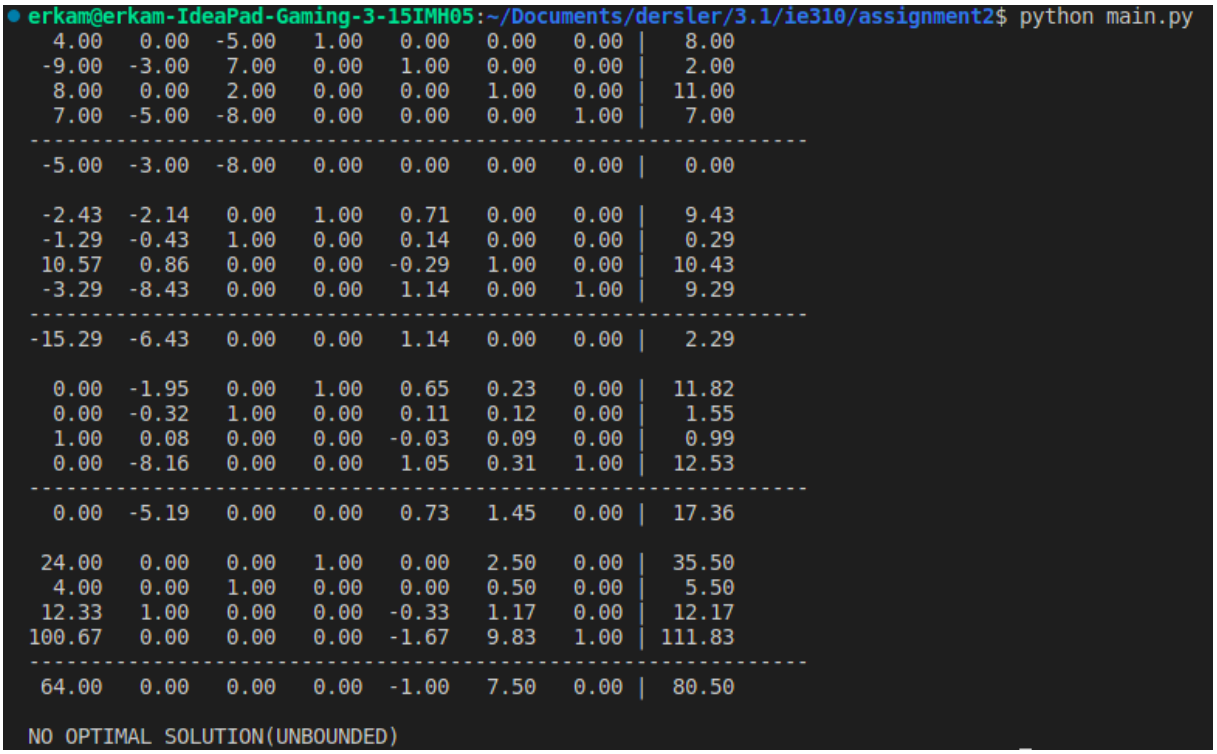
z0, is the result obtained by the current solution.

# Calculated Results

- Data1.txt




- Data2.txt

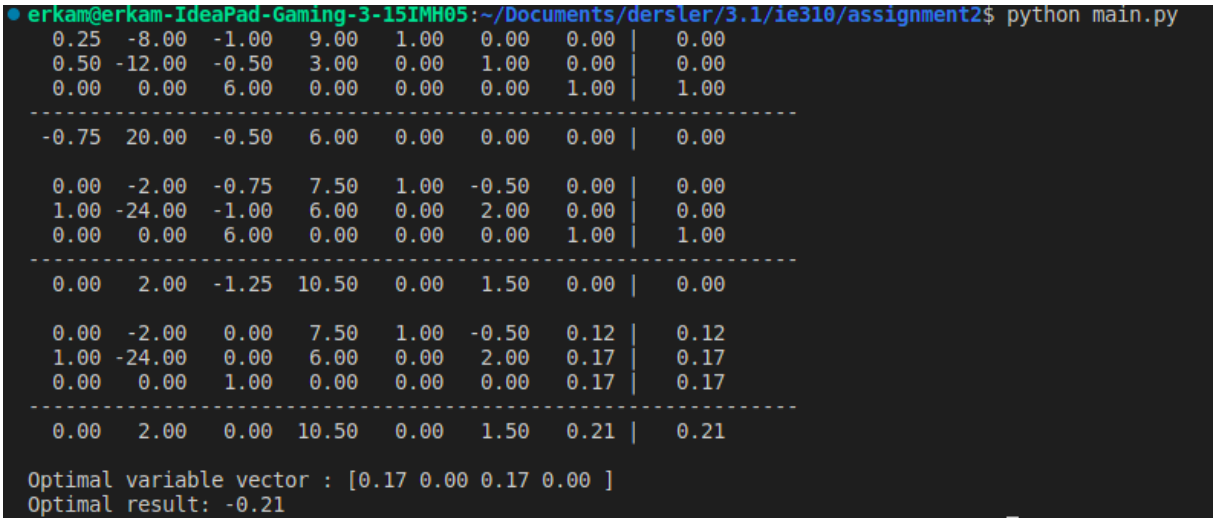


- Data3.txt

```
erkam@erkam-IdeaPad-Gaming-3-15IMH05:~/Documents/dersler/3.1/ie310/assignment2$ python main.py
  1.00   2.00   3.00   7.00  -3.00   7.00   9.00   1.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00 |   3.00
  4.00  -1.00  -8.00   6.00   4.00  -4.00   9.00   6.00   0.00   1.00   0.00   0.00   0.00   0.00   0.00 |   8.00
  9.00   7.00  -2.00   4.00  -3.00  -7.00   4.00   3.00   0.00   0.00   1.00   0.00   0.00   0.00   0.00 |   8.00
  1.00  -2.00   9.00  -8.00  -4.00  -3.00  -5.00  -1.00   0.00   0.00   0.00   1.00   0.00   0.00   0.00 |   8.00
  7.00  -9.00   0.00  -2.00   8.00   2.00  -1.00   0.00   0.00   0.00   0.00   0.00   1.00   0.00   0.00 |   4.00
  8.00  -3.00   8.00   5.00  -3.00  -9.00   4.00   2.00   0.00   0.00   0.00   0.00   0.00   1.00   0.00 |   7.00
  7.00   4.00   8.00   5.00  -5.00  -9.00  -8.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   1.00 |   3.00
-------------------------------------------------------------------------------------------------------------------------------
  7.00  -3.00  -6.00   5.00   0.00  -1.00  -4.00  -4.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00 |   0.00

 -1.62   0.50   0.00   5.12  -1.12  10.38  12.00   0.62   1.00   0.00   0.00   0.00   0.00   0.00  -0.38 |   1.88
 11.00   3.00   0.00  11.00  -1.00 -13.00   1.00   7.00   0.00   1.00   0.00   0.00   0.00   0.00   1.00 |  11.00
 10.75   8.00   0.00   5.25  -4.25  -9.25   2.00   3.25   0.00   0.00   1.00   0.00   0.00   0.00   0.25 |   8.75
 -6.88  -6.50   0.00 -13.62   1.62   7.12   4.00  -2.12   0.00   0.00   0.00   1.00   0.00   0.00  -1.12 |   4.62
  7.00  -9.00   0.00  -2.00   8.00   2.00  -1.00   0.00   0.00   0.00   0.00   0.00   1.00   0.00   0.00 |   4.00
  1.00  -7.00   0.00   0.00   2.00   0.00  12.00   1.00   0.00   0.00   0.00   0.00   0.00   1.00  -1.00 |   4.00
  0.88   0.50   1.00   0.62  -0.62  -1.12  -1.00   0.12   0.00   0.00   0.00   0.00   0.00   0.00   0.12 |   0.38
-------------------------------------------------------------------------------------------------------------------------------
 12.25   0.00   0.00   8.75  -3.75  -7.75 -10.00  -3.25   0.00   0.00   0.00   0.00   0.00   0.00   0.75 |   2.25

 -0.14   0.04   0.00   0.43  -0.09   0.86   1.00   0.05   0.08   0.00   0.00   0.00   0.00   0.00  -0.03 |   0.16
 11.14   2.96   0.00  10.57  -0.91 -13.86   0.00   6.95  -0.08   1.00   0.00   0.00   0.00   0.00   1.03 |  10.84
 11.02   7.92   0.00   4.40  -4.06 -10.98   0.00   3.15  -0.17   0.00   1.00   0.00   0.00   0.00   0.31 |   8.44
 -6.33  -6.67   0.00 -15.33   2.00   3.67   0.00  -2.33  -0.33   0.00   0.00   1.00   0.00   0.00  -1.00 |   4.00
  6.86  -8.96   0.00  -1.57   7.91   2.86   0.00   0.05   0.08   0.00   0.00   0.00   1.00   0.00  -0.03 |   4.16
  2.62  -7.50   0.00  -5.12   3.12 -10.38   0.00   0.38  -1.00   0.00   0.00   0.00   0.00   1.00  -0.62 |   2.12
  0.74   0.54   1.00   1.05  -0.72  -0.26   0.00   0.18   0.08   0.00   0.00   0.00   0.00   0.00   0.09 |   0.53
-------------------------------------------------------------------------------------------------------------------------------
 10.90   0.42   0.00  13.02  -4.69   0.90   0.00  -2.73   0.83   0.00   0.00   0.00   0.00   0.00   0.44 |   3.81

 -0.05  -0.06   0.00   0.41   0.00   0.90   1.00   0.05   0.08   0.00   0.00   0.00   0.01   0.00  -0.03 |   0.21
 11.92   1.93   0.00  10.39   0.00 -13.54   0.00   6.95  -0.07   1.00   0.00   0.00   0.11   0.00   1.03 |  11.32
 14.55   3.31   0.00   3.59   0.00  -9.51   0.00   3.17  -0.12   0.00   1.00   0.00   0.51   0.00   0.30 |  10.57
 -8.07  -4.40   0.00 -14.94   0.00   2.94   0.00  -2.35  -0.35   0.00   0.00   1.00  -0.25   0.00  -0.99 |   2.95
  0.87  -1.13   0.00  -0.20   1.00   0.36   0.00   0.01   0.01   0.00   0.00   0.00   0.13   0.00  -0.00 |   0.53
 -0.09  -3.96   0.00  -4.50   0.00 -11.51   0.00   0.35  -1.03   0.00   0.00   0.00  -0.40   1.00  -0.61 |   0.48
  1.36  -0.27   1.00   0.91   0.00  -0.00   0.00   0.18   0.09   0.00   0.00   0.00   0.09   0.00   0.09 |   0.91
-------------------------------------------------------------------------------------------------------------------------------
 14.97  -4.89   0.00  12.09  -0.00   2.59   0.00  -2.70   0.88   0.00   0.00   0.00   0.59   0.00   0.42 |   6.28

  0.23   0.00   0.00   0.48   0.00   0.71   1.00   0.11   0.08   0.00   0.02   0.00   0.02   0.00  -0.03 |   0.41
  3.44   0.00   0.00   8.30   0.00  -7.99   0.00   5.10  -0.00   1.00  -0.58   0.00  -0.18   0.00   0.85 |   5.16
  4.39   1.00   0.00   1.08   0.00  -2.87   0.00   0.96  -0.04   0.00   0.30   0.00   0.16   0.00   0.09 |   3.19
 11.25  -0.00   0.00 -10.17   0.00  -9.68   0.00   1.87  -0.52   0.00   1.33   1.00   0.43   0.00  -0.60 |  16.99
  5.84   0.00   0.00   1.03   1.00  -2.89   0.00   1.09  -0.03   0.00   0.34   0.00   0.30   0.00   0.10 |   4.14
 17.29   0.00   0.00  -0.22   0.00 -22.87   0.00   4.15  -1.18   0.00   1.19   0.00   0.22   1.00  -0.26 |  13.12
  2.56   0.00   1.00   1.20   0.00  -0.78   0.00   0.44   0.08   0.00   0.08   0.00   0.13   0.00   0.12 |   1.78
-------------------------------------------------------------------------------------------------------------------------------
 36.46   0.00   0.00  17.39   0.00 -11.45   0.00   1.99   0.70   0.00   1.48   0.00   1.35   0.00   0.86 |  21.89
```

```
  0.32   0.00   0.00   0.67   0.00   1.00   1.40   0.16   0.11   0.00   0.03   0.00   0.03   0.00  -0.04 |   0.58
  6.01   0.00   0.00  13.66   0.00   0.00  11.21   6.39   0.92   1.00  -0.36   0.00   0.06   0.00   0.57 |   9.77
  5.31   1.00   0.00   3.01   0.00   0.00   4.02   1.42   0.29   0.00   0.38   0.00   0.24   0.00  -0.01 |   4.85
 14.37   0.00   0.00  -3.68   0.00   0.00  13.58   3.42   0.59   0.00   1.59   1.00   0.73   0.00  -0.95 |  22.58
  6.77   0.00   0.00   2.96   1.00   0.00   4.05   1.56   0.30   0.00   0.42   0.00   0.39   0.00  -0.01 |   5.81
 24.65   0.00   0.00  15.12   0.00   0.00  32.06   7.82   1.44   0.00   1.82   0.00   0.92   1.00  -1.09 |  26.31
  2.81   0.00   1.00   1.73   0.00  -0.00   1.10   0.57   0.17   0.00   0.10   0.00   0.16   0.00   0.09 |   2.23
-------------------------------------------------------------------------------------------------------------------------------
 40.14   0.00   0.00  25.07   0.00   0.00  16.05   3.83   2.01   0.00   1.79   0.00   1.70   0.00   0.44 |  28.50

Optimal variable vector : [0.00 4.85 2.23 0.00 5.81 0.58 0.00 0.00 ]
Optimal result: -28.50
```

- Problem2 Solution




# Implementation

Firstly, the input file is parsed, and A matrix, b and c vectors are obtained from the input file. Since A, b, c variables are multiplied with each other constantly, they are stored in matrix style. Which is implemented via the two-dimensional array in the python.

Secondly, the slack variables are added to the problem. To start advancing with the Simplex algorithm, the problem must be in canonical form. This means that right hand side variables should be positive, the constraints should be equal to, and the combination of basic variable columns should be an identity matrix. The given input files already have non-negative right-hand sides. To achieve canonical form each constraint is extended with a slack variable which turns all constraints to equal to. After this change, the updated A matrix and the c vector is ready for the Simplex algorithm. And the initial base is composed of these slack variables. This means that the Simplex algorithm starts with a feasible solution.

The Simplex algorithm always tries to improve the optimal result with swapping indexes with each other while staying in the feasible region. So, the algorithm starts with choosing an incoming index looking for the minimum value in the c vector. If there are no negative coefficients, the Simplex algorithm can not choose an incoming index because choosing a non-negative coefficient in the c vector will not improve the current tableu. Therefore, the Simplex algorithm exit the loop because the current tableu is optimal.

After choosing an incoming index in the base, the algorithm starts searching for an outgoing index. The Simplex algorithm must remain in the feasible region, so a ratio test is applied using the column of the incoming index in the A matrix and the b vector. The minimum non-negative ratio is chosen to be the outgoing index otherwise next solution won't be feasible. If there are no non-negative ratio, this means that the current incoming variable is not bounded by a constraint, therefore, the problem is unbounded and there is no optimal solution.

Later the new base in constructed with removing the outgoing index and placing the incoming index. Then the new A, b, c and z0 variables are calculated from this new base and the Simplex algorithm returns to the beginning. The algorithm stops when it reached the optimal solution, or it has decided that the current problem is unbounded.

# Problem 2

In the given problem, the A matrix already contained the slack variables. To be able to run this problem in our algorithm we removed the first 3 variables and gave the remaining matrix to the algorithm because we expected the algorithm to add those slack variables in the end. The remaining parts are the same.