МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

**Лабораторная работа №4**

Выполнил:

Студент группы ВМО 31

Оганесьянц К.П.

2021г

**Цель работы:** нахождение оптимальных решений задач линейного программирования с использованием симплекс-метода.

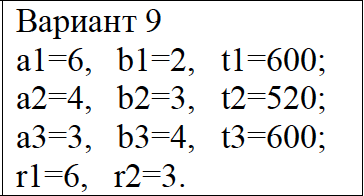
**Задание**:

1. Составить математическую модель задачи.

2. Решить ЗЛП симплекс-методом.

Для производства двух моделей автомобилей А и В используется три типа технологического оборудования. На производство одного автомобиля модели А, на оборудовании первого типа используется a1 часов, на оборудо-вании второго типа – a2 часов, третьего типа – a3 часов. На производство од-ного автомобиля модели В оборудованием первого типа используется b1 ча-сов, оборудованием второго типа – b2 часов, третьего типа – b3 часов. На изготовление всех изделий администрация завода может предоста- вить оборудование первого типа не более, чем на t 1 часов, оборудование вто-рого типа не более, чем на t 2 часов, оборудование третьего типа не более, чем на t 3 часов. Прибыль от реализации одного автомобиля марки А составляет r 1 тыс. д. е. , марки В – r 2 тыс. д. е.

Составить план выпуска автомобилей, обеспечивающий максимальную прибыль.



Приложение №1: Программная реализация

Файл №1: simplex.py – основной файл с математическими операциями, решающими данную задачу:

import math

import numpy as np

def get\_pivot\_position(tableau):

z = tableau[-1]

column = next(i for i, x in enumerate(z[:-1]) if x > 0)

restrictions = []

for eq in tableau[:-1]:

el = eq[column]

restrictions.append(math.inf if el <= 0 else eq[-1] / el)

row = restrictions.index(min(restrictions))

return row, column

def pivot\_step(tableau, pivot\_position):

new\_tableau = [[] for eq in tableau]

i, j = pivot\_position

pivot\_value = tableau[i][j]

new\_tableau[i] = np.array(tableau[i]) / pivot\_value

for eq\_i, eq in enumerate(tableau):

if eq\_i != i:

multiplier = np.array(new\_tableau[i]) \* tableau[eq\_i][j]

new\_tableau[eq\_i] = np.array(tableau[eq\_i]) - multiplier

return new\_tableau

def is\_basic(column):

return sum(column) == 1 and len([c for c in column if c == 0]) == len(column) - 1

def get\_solution(tableau):

columns = np.array(tableau).T

solutions = []

for column in columns[:-1]:

solution = 0

if is\_basic(column):

one\_index = column.tolist().index(1)

solution = columns[-1][one\_index]

solutions.append(solution)

return solutions

def simplex(c, A, b):

tableau = to\_tableau(c, A, b)

while can\_be\_improved(tableau):

pivot\_position = get\_pivot\_position(tableau)

tableau = pivot\_step(tableau, pivot\_position)

return get\_solution(tableau)

def to\_tableau(c, A, b):

xb = [eq + [x] for eq, x in zip(A, b)]

z = c + [0]

return xb + [z]

def can\_be\_improved(tableau):

z = tableau[-1]

return any(x > 0 for x in z[:-1])

print("Введите прибыль через запятую")

tmp\_c = str(input()).replace(" ","").split(',')

c=list(map(int, tmp\_c))

# c = [40, 50, 30, 20,0,0, 0]

print()

print("Введите вектор b:")

tmp\_c = str(input()).replace(" ","").split(',')

b=list(map(int, tmp\_c))

A=[]

print("Введите матрицу A")

for i in range(0,len(b)):

print("Введи строку x-ов через ','")

tmp\_c = str(input()).replace(" ","").split(',')

tmpVec=list(map(int, tmp\_c))

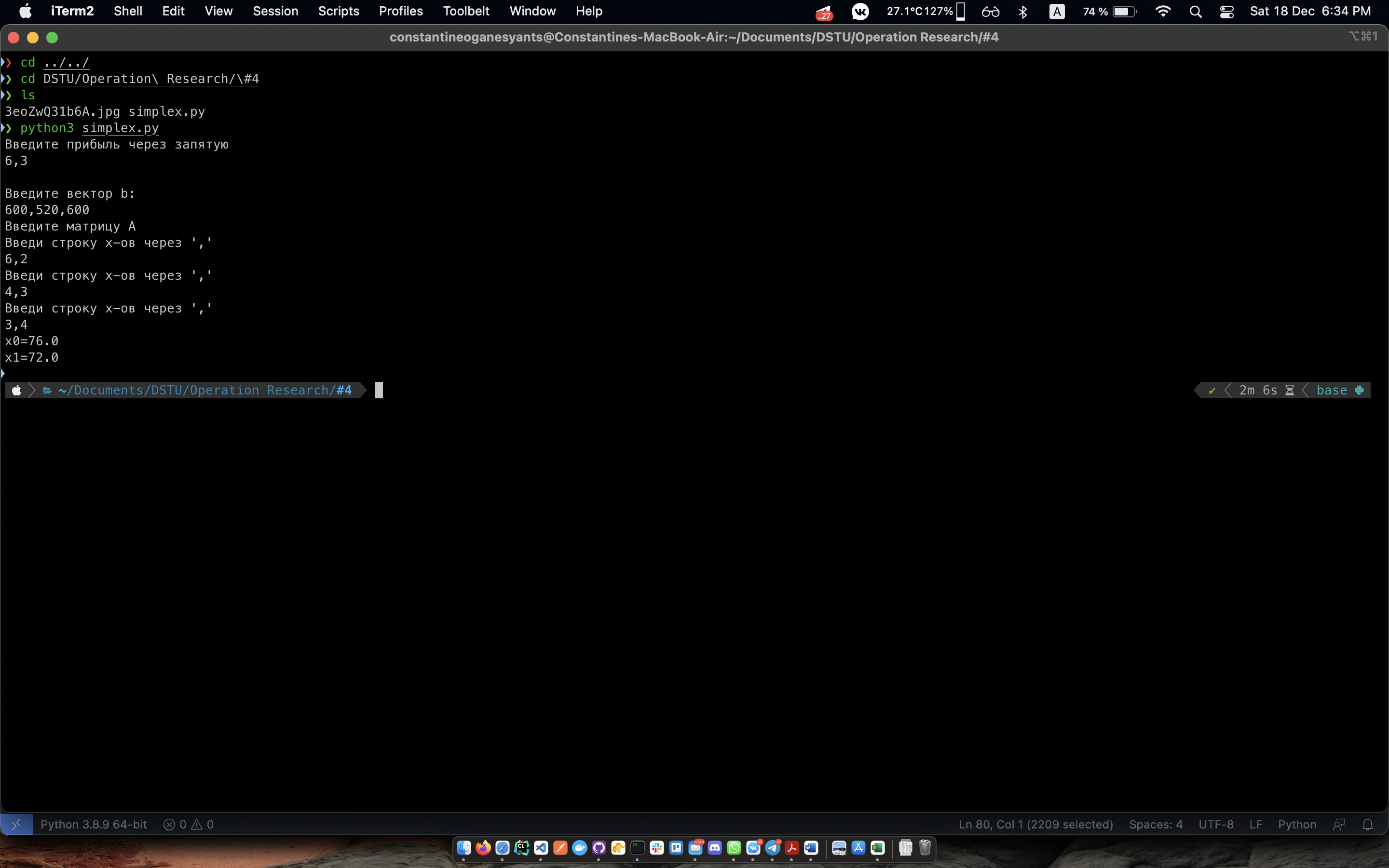
A.append(tmpVec)

solution = simplex(c, A, b)

for i in range(0, len(solution)):

print(f"x{i}={solution[i]}")

Приложение №2: Вывод реализованного программного средства:



Приложение №3: реализация данной задачи в среде MathCad:

