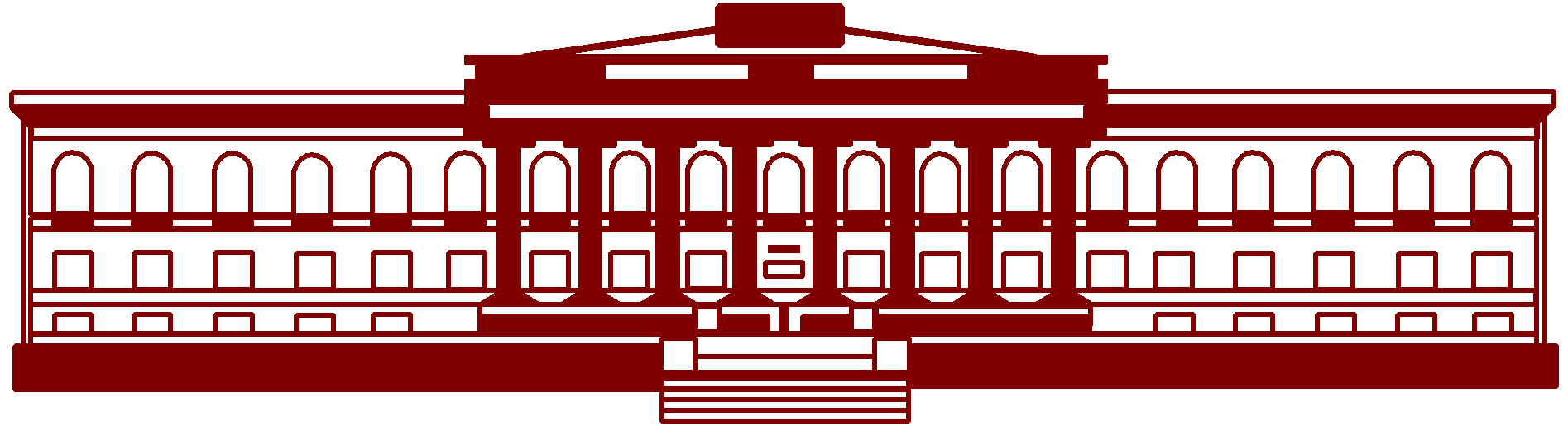
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**імені ТАРАСА ШЕВЧЕНКА**

****

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №8**

**з курсу**

**«Системний аналіз та теорія прийняття рішень»**

*Студентa 3 курсу*

*групи ПП-32*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

*Кондратова Івана Андрійовича*

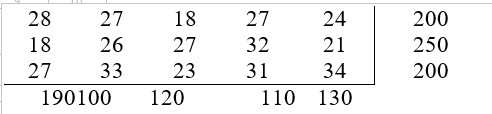
*Викладач:*

Плескач В.Л.

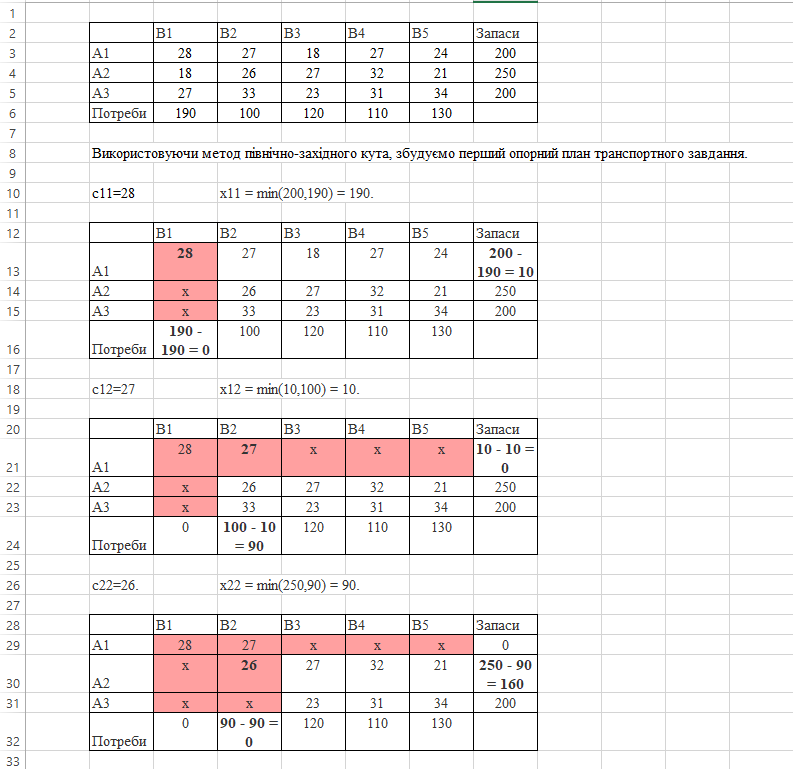
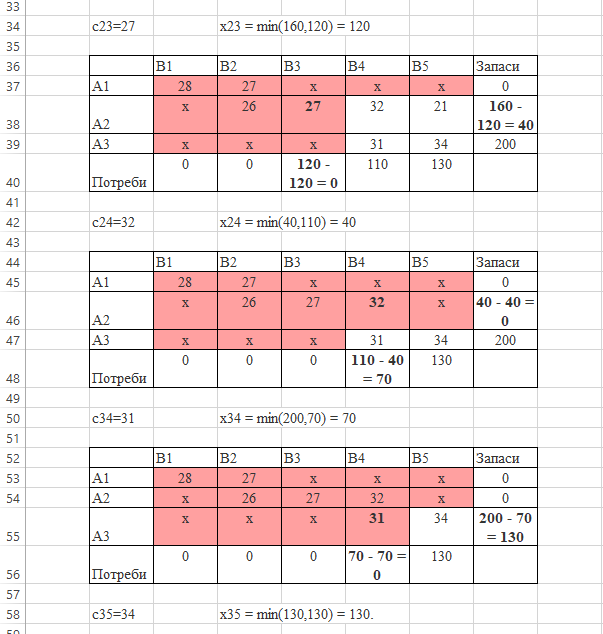
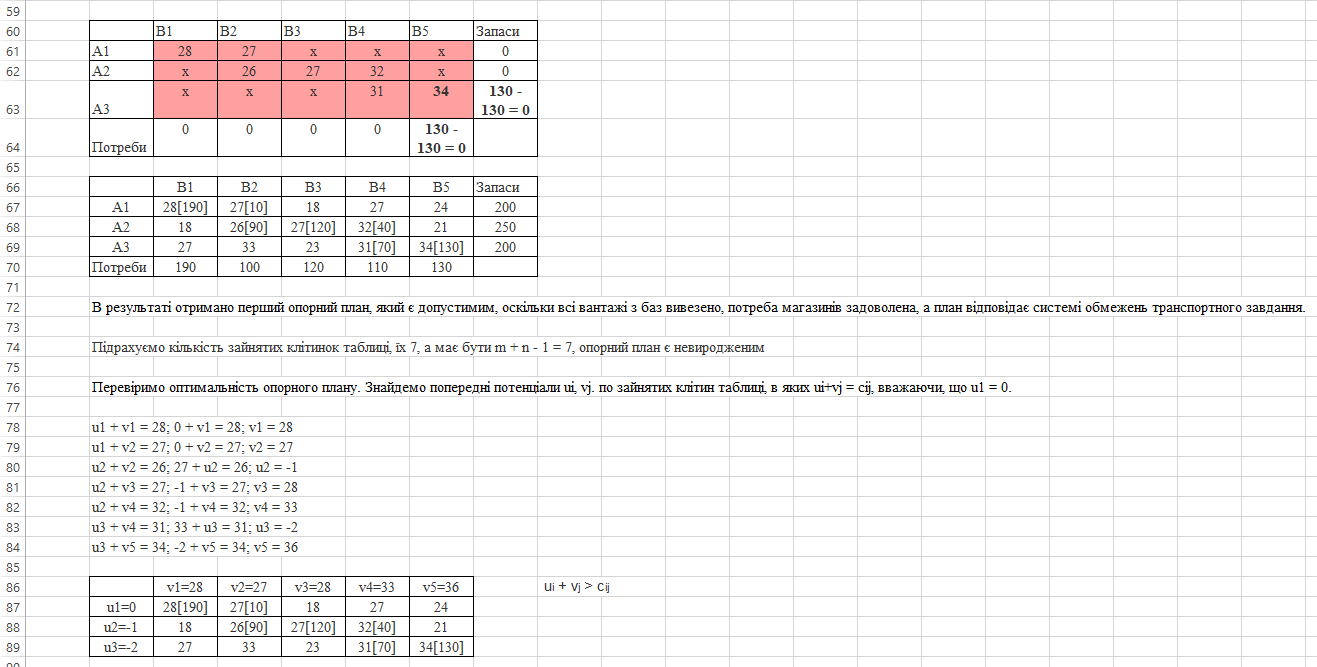
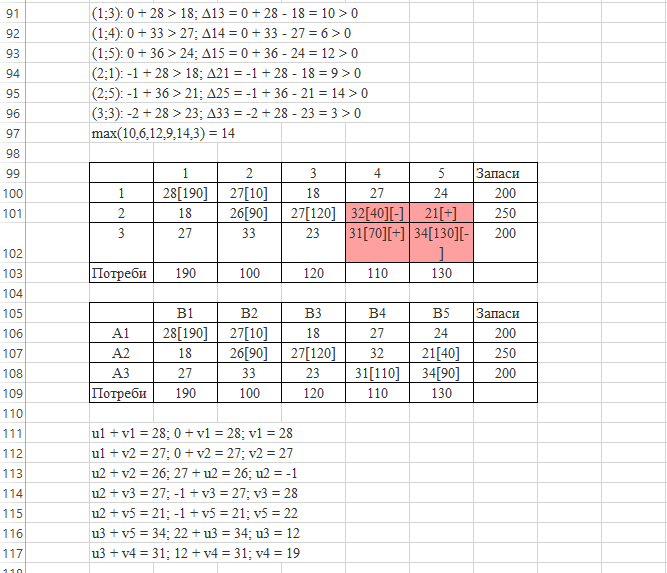
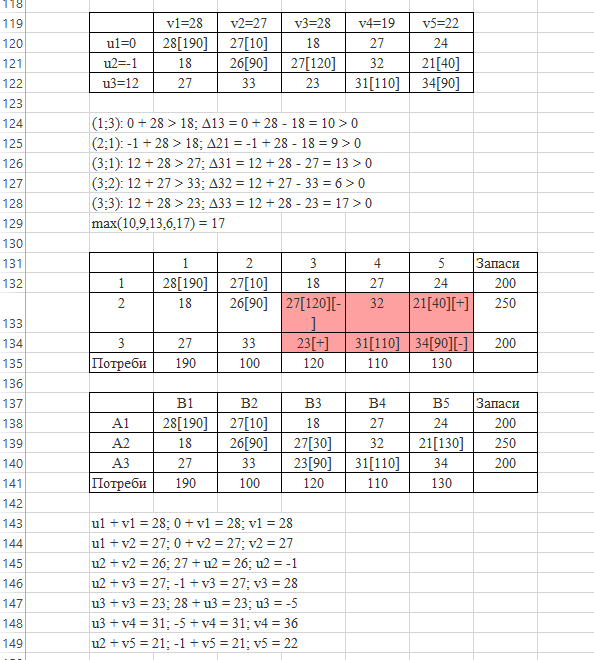
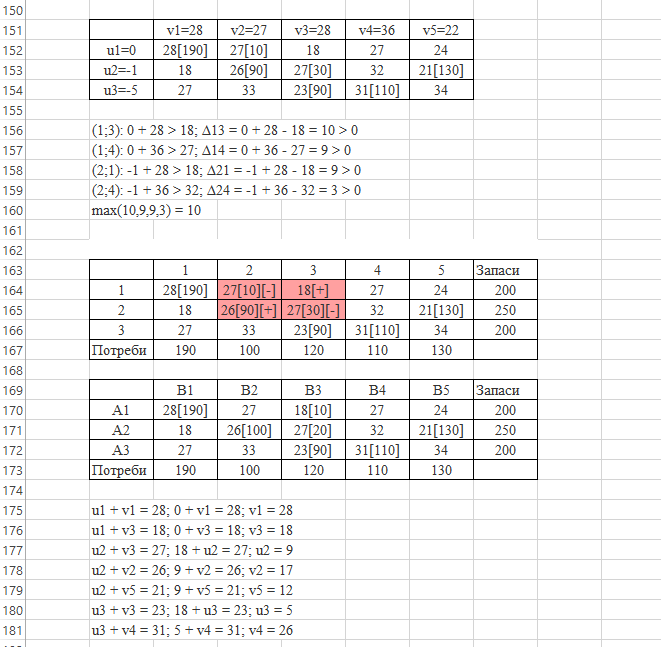
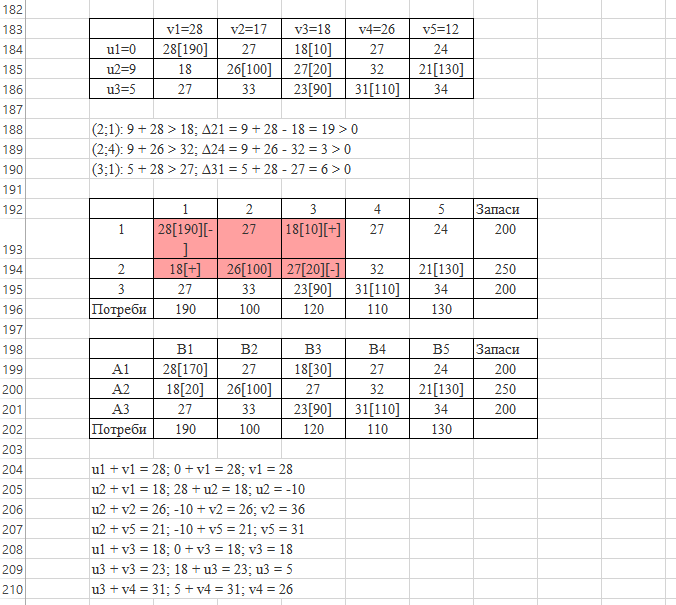
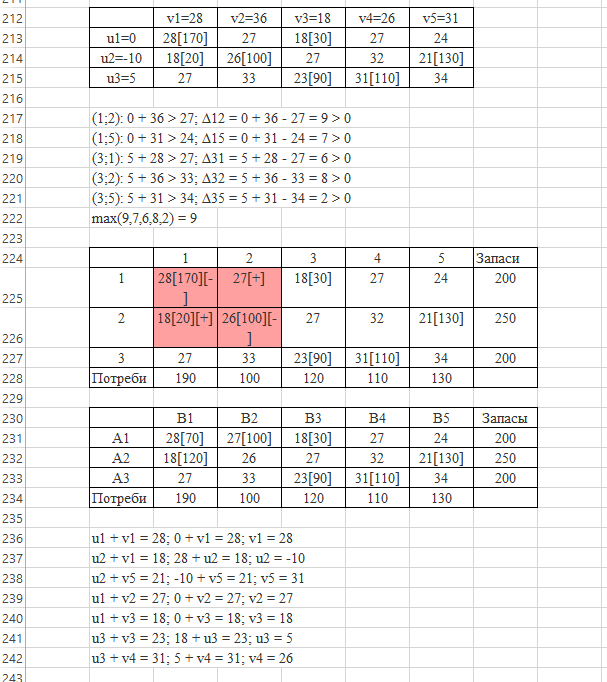
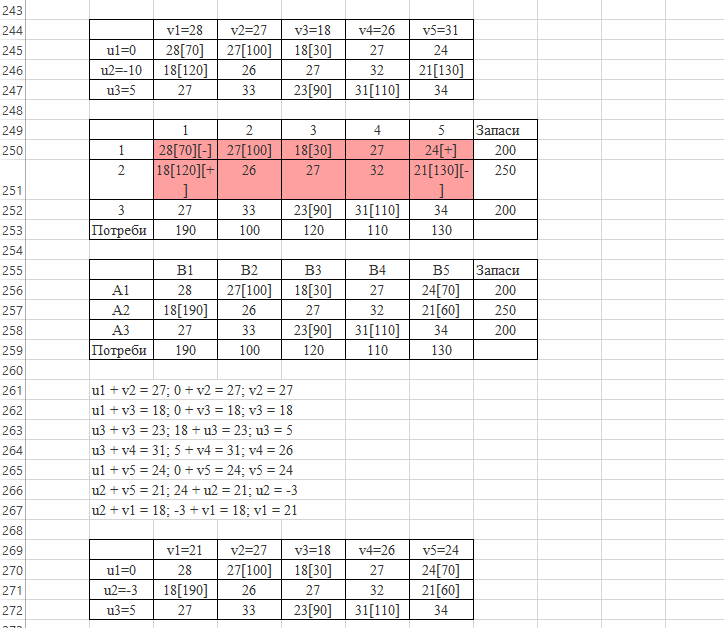
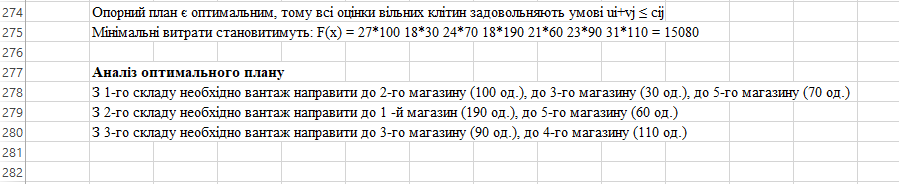
**Київ – 2023**

**Тема:** транспортна задача

ЗАДАЧА 6.



**Розв’язання в Excel**

**Розв’язання в Python**

import numpy as np

def nw\_angle(supply\_arr, demand\_arr):

    rows = len(supply\_arr)

    cols = len(demand\_arr)

    result = np.zeros((rows, cols))

    i = 0

    j = 0

    while i < rows and j < cols:

        quantity = min(supply\_arr[i], demand\_arr[j])

        result[i, j] = quantity

        supply\_arr[i] -= quantity

        demand\_arr[j] -= quantity

        if supply\_arr[i] == 0:

            i += 1

        if demand\_arr[j] == 0:

            j += 1

    return result

def main():

    supply\_arr = np.array([200,250,200])

    demand\_arr = np.array([190, 100,120,110,130])

    cost\_matrix = np.array([

        [28,27, 18, 27, 24],

        [18,26, 27, 32, 21],

        [27,33, 23, 31, 34],

        ])

    nw\_result = nw\_angle(supply\_arr, demand\_arr)

    result = np.sum(nw\_result \* cost\_matrix)

    print(nw\_result)

    print(result)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

from pulp import \*

import numpy as np

import pandas as pd

# Постачання

supply = [200, 250, 200]

suppliers = range(len(supply))

# Попит

demand = [190,100,120,110,130]

consumers = range(len(demand))

# Вартість перевезення

costs = [[28, 27,18,27, 24], [18,26,27,32,21], [27,33,23,31,34]]

# Дані для задачі

costs = makeDict([suppliers, consumers], costs, 0)

prob = LpProblem("Transportation\_Problem", LpMinimize)

# Шляхи перевезення

routes = [(i, j) for i in suppliers for j in consumers]

# Змінні для шляхів перевезення

vars = LpVariable.dicts("Route", (suppliers, consumers), 0, None, LpInteger)

# Функція мінімізації

prob += (

    lpSum([vars[w][b] \* costs[w][b] for (w, b) in routes]),

    "Sum\_of\_Transporting\_Costs",

)

# Максимальні обмеження постачання

for s in suppliers:

    prob += (

        lpSum([vars[s][b] for b in consumers]) <= supply[s],

        "Sum\_of\_Products\_out\_of\_supplier\_%s" % s,

    )

# Мінімальні обмеження попиту

for c in consumers:

    prob += (

        lpSum([vars[w][c] for w in suppliers]) >= demand[c],

        "Sum\_of\_Products\_into\_consumer%s" % c,

    )

# Вирішення задачі за допомогою PuLP

prob.solve()

results = [v.varValue for v in prob.variables()]

print("Відповідь:")

print(pd.DataFrame(np.array(results).reshape(3, 5)))

print("Загальна вартість:", value(prob.objective))

**Висновок:** виконуючи цю лабораторну роботу, я навчився розв’язувати транспортні задачі методами північно-західного кута та потенціалів