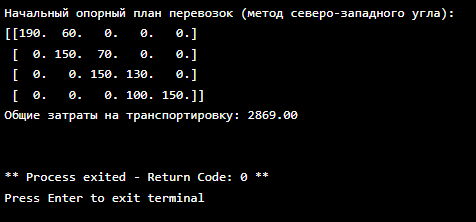
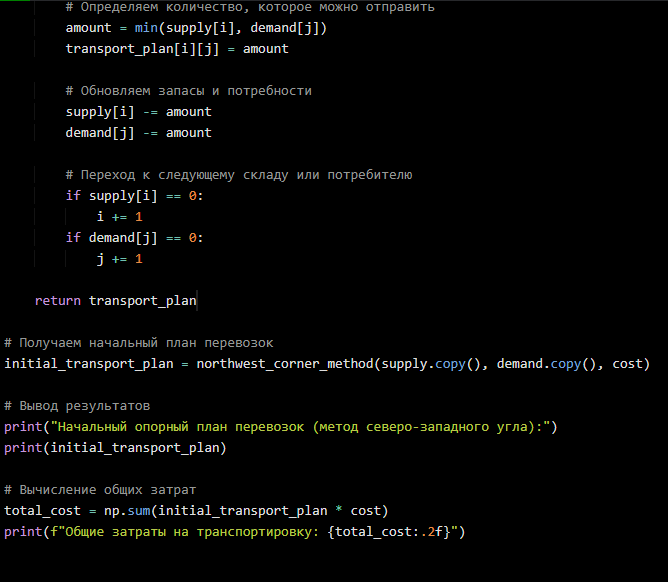
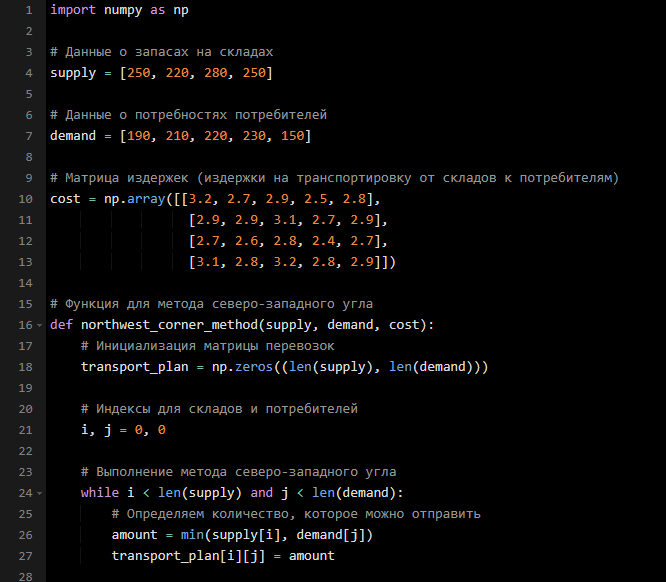
**Задание 2.**

****

**Листинг:**

**import numpy as np**

**# Данные о запасах на складах**

**supply = [250, 220, 280, 250]**

**# Данные о потребностях потребителей**

**demand = [190, 210, 220, 230, 150]**

**# Матрица издержек (издержки на транспортировку от складов к потребителям)**

**cost = np.array([[3.2, 2.7, 2.9, 2.5, 2.8],**

**[2.9, 2.9, 3.1, 2.7, 2.9],**

**[2.7, 2.6, 2.8, 2.4, 2.7],**

**[3.1, 2.8, 3.2, 2.8, 2.9]])**

**# Функция для метода северо-западного угла**

**def northwest\_corner\_method(supply, demand, cost):**

**# Инициализация матрицы перевозок**

**transport\_plan = np.zeros((len(supply), len(demand)))**

**# Индексы для складов и потребителей**

**i, j = 0, 0**

**# Выполнение метода северо-западного угла**

**while i < len(supply) and j < len(demand):**

**# Определяем количество, которое можно отправить**

**amount = min(supply[i], demand[j])**

**transport\_plan[i][j] = amount**

**# Обновляем запасы и потребности**

**supply[i] -= amount**

**demand[j] -= amount**

**# Переход к следующему складу или потребителю**

**if supply[i] == 0:**

**i += 1**

**if demand[j] == 0:**

**j += 1**

**return transport\_plan**

**# Получаем начальный план перевозок**

**initial\_transport\_plan = northwest\_corner\_method(supply.copy(), demand.copy(), cost)**

**# Вывод результатов**

**print("Начальный опорный план перевозок (метод северо-западного угла):")**

**print(initial\_transport\_plan)**

**# Вычисление общих затрат**

**total\_cost = np.sum(initial\_transport\_plan \* cost)**

**print(f"Общие затраты на транспортировку: {total\_cost:.2f}")**