**Содержание**

[Задание 2](#_Toc57920040)

[Математическая формулировка задачи 3](#_Toc57920041)

[Текст программы на языке С++ 4](#_Toc57920043)

[Список литературы 5](#_Toc57920048)

# **Задание**

Имеется некоторое количество авиарейсов между городами мира, для каждого известна стоимость. Стоимость перелёта из A в B может быть не равна стоимости перелёта из B в A. Найти маршруты минимальной стоимости между городами (возможно, с пересадками).

**Формат входных данных**

Во входном файле записано сначала число N (1<=N<=100), определявшее количество рассматриваемых городов. Затем идет число M - количество авиарейсов между городами мира, далее идет описание самих дорог. Каждое сообщение между городами задается тремя числами - номерами городов и стоимостью перелета.

**Формат выходных данных**

На экран выведите числа - суммарные стоимости маршрутов с указанием номеров городов или -1, если добраться невозможно.

# **Математическая формулировка задачи**

***Алгоритм Флойда.***

# Обозначим длину кратчайшего пути между вершинами u и v, содержащего, помимо u и v, только вершины из множества {1..i}как d(i)uv , d(0)uv=ωuv. На каждом шаге алгоритма, мы будем брать очередную вершину (пусть её номер — i) и для всех пар вершин u и v вычислять d(i)uv=min(d(i−1)uv,d(i−1)ui+d(i−1)iv). То есть, если кратчайший путь из u в v, содержащий только вершины из множества {1..i}, проходит через вершину i, то кратчайшим путем из u в v является кратчайший путь из u в i, объединенный с кратчайшим путем из i в v. В противном случае, когда этот путь не содержит вершины i, кратчайший путь из u в v, содержащий только вершины из множества {1..i} является кратчайшим путем из u в v, содержащим только вершины из множества {1..i−1}.

# **Текст программы на языке Python**

N = int(input())

M = int(input())

a = [[float('inf')]\*N for i in range(N)]

for i in range(N):

a[i][i] = 0

for i in range(M):

b = input().split()

a[int(b[0])-1][int(b[1])-1] = int(b[2])

for i in range(N):

for ii in range(N):

for iii in range(N):

a[ii][iii] = min(a[ii][iii], a[ii][i] + a[i][iii])

for i in range(N):

for j in range(N):

if a[i][j] != float('inf'):

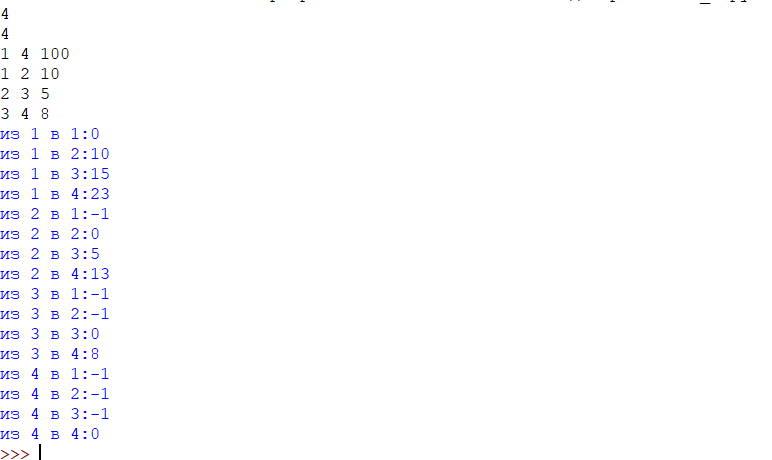
print(f"из {(i+1)} в {(j+1)}:{a[i][j]}")

else:

print(f"из {(i+1)} в {(j+1)}:-1")

# **Разработка интерфейса пользователя**

На рисунке 2 представлено программа, её работа осуществляется путем ввода входных данных.



*Рисунок 2 – Работа программы*

# **Список литературы**

1. Викиконспекты – Алгоритм Флойда([https://neerc.ifmo.ru/wiki/index.php) 15.05.2023](https://neerc.ifmo.ru/wiki/index.php)%2015.05.2023)