/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* OPL 12.8.0.0 Model

\* Author: Rodrigo Rosa

\* Creation Date: 21/05/2019 at 19:09:16

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//número de clientes

int n = ...;

//número de veículos

int k = ...;

//capacidade do veiculo

float Q = ...;

//tempo de atendimento

//float D = ...;

//conjunto de clientes

range C = 1 .. n;

//conjunto de clientes

range C1 = 1 .. n+1;

//conjunto de clientes

range C0 = 0 .. n;

//conjunto de clientes mais depósitos

range N = 0 .. n+1;

//conjunto de veículos

range K = 1 .. k;

//matriz de tempo entre nós

float t[i in N][j in N] = ...;

//matriz de distância entre nós

float c[i in N][j in N] = ...;

//vetor de demanda

float d[i in C] = ...;

//vetor de demanda

float a[i in C1] = ...;

//vetor de demanda

float b[i in C1] = ...;

int M = 99999;

//variavel de decisao-se = 0 entao nao e pecorrido o arco i-j pelo veiculo k,

//se = 1 entao o veiculo k percorre o arco i-j

dvar int x[i in N][j in N][k in K];

dvar float s[i in N][k in K];

//funcao objetivo

minimize sum(k in K, i in N, j in N) (c[i][j] \* x[i][j][k]);

subject to {

forall (i in C) {

const01: sum(k in K, j in N) x[i][j][k] == 1;}

forall (k in K) {

const02: sum(i in C, j in N) (d[i ]\* x[i][j][k]) <= Q ;}

forall (k in K) {

const03: s[0,k] <= 0;}

forall (k in K) {

const03a: s[n+1,k] <= b[n+1];}

// const03a: s[n+1,k] <= D;}

forall (k in K) {

const04: sum(j in N) x[0][j][k] == 1;}

forall (k in K, h in C) {

const05: sum(i in N) x[i][h][k] - sum(j in N) x[h][j][k] == 0;}

forall (k in K) {

const06: sum(i in N) x[i][n+1][k] == 1;}

forall (i in N, j in N, k in K) {

const07: 0 <= x[i][j][k] <= 1;}

forall (i in C1, k in K) {

const08: a[i] <= s[i][k] <= b[i];}

forall (i in C0, j in C1, k in K) {

if (i != j) {

const09: s[j][k] >= s[i][k] + t[i][j] - (1 - x[i][j][k])\*M ; }}

forall (i in N, k in K) {

const10: s[i][k] >= 0;}

}

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

Aqui comeca o dat

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* OPL 12.8.0.0 Data

\* Author: Rodrigo Rosa

\* Creation Date: 21/05/2019 at 19:09:16

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Definição dos dados (constantes)

//numero de clientes (int)

n = 9;

//número de veículos da frota (int)

k = 3;

//k = 1;

// capacidade do veículo (int)

Q = 60;

//Q = 140;

c = [

[9999.99 38.47 55.95 18.03 49.09 51.48 38.21 38.42 52.39 14.87 0.00]

[9999.99 9999.99 31.78 40.80 45.54 89.90 62.51 7.21 85.91 53.23 38.47]

[9999.99 9999.99 9999.99 66.22 77.28 101.86 59.62 25.18 108.23 67.80 55.95]

[9999.99 9999.99 9999.99 9999.99 33.24 57.31 55.90 43.83 45.28 26.00 18.03]

[9999.99 9999.99 9999.99 9999.99 9999.99 88.20 87.01 52.17 65.46 59.24 49.09]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 50.70 89.19 37.48 36.67 51.48]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 58.03 74.33 34.01 38.21]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 88.23 52.70 38.42]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 43.57 52.39]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 14.87]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99]

];

t = [

[9999.99 46.16 67.14 21.63 58.91 61.77 45.85 46.10 62.87 17.84 0.00]

[9999.99 9999.99 38.14 48.97 54.65 107.88 75.02 8.65 103.10 63.87 46.16]

[9999.99 38.14 9999.99 79.46 92.73 122.24 71.54 30.22 129.87 81.36 67.14]

[9999.99 48.97 79.46 9999.99 39.89 68.78 67.08 52.60 54.33 31.20 21.63]

[9999.99 54.65 92.73 39.89 9999.99 105.85 104.41 62.61 78.55 71.08 58.91]

[9999.99 107.88 122.24 68.78 105.85 9999.99 60.83 107.02 44.98 44.01 61.77]

[9999.99 75.02 71.54 67.08 104.41 60.83 9999.99 69.64 89.20 40.82 45.85]

[9999.99 8.65 30.22 52.60 62.61 107.02 69.64 9999.99 105.88 63.24 46.10]

[9999.99 103.10 129.87 54.33 78.55 44.98 89.20 105.88 9999.99 52.28 62.87]

[9999.99 63.87 81.36 31.20 71.08 44.01 40.82 63.24 52.28 9999.99 17.84]

[9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99 9999.99]

];

// Demanda dos clientes (int)

d = [11 35 2 9 3 18 8 10 11];

// Início da janela de tempo cliente i (int)

a = [45 11 25 20 15 50 10 40 10 0];

// Fim da janela de tempo cliente (int)

b = [70 145 40 100 80 190 110 190 45 400];

// Limite de tempo para rota de cada veículo

D = 400;