// --------------------------------------------------------------------------

// Licensed Materials - Property of IBM

//

// 5725-A06 5725-A29 5724-Y48 5724-Y49 5724-Y54 5724-Y55

// Copyright IBM Corporation 1998, 2020. All Rights Reserved.

//

// Note to U.S. Government Users Restricted Rights:

// Use, duplication or disclosure restricted by GSA ADP Schedule

// Contract with IBM Corp.

// --------------------------------------------------------------------------

using CP;

int SizeSquare = 112;

int NbSquares = 21;

execute{

}

range Squares = 1..NbSquares;

int Size[Squares] = [50,42,37,35,33,29,27,25,24,19,18,17,16,15,11,9,8,7,6,4,2];

// 21 square to fit into a bigger square

// each square consume space of the square

// we can scheduling each square as interval from 0 to bigger sqaure's size'

dvar interval x[s in Squares] in 0..SizeSquare size Size[s];

dvar interval y[s in Squares] in 0..SizeSquare size Size[s];

cumulFunction rx = sum(s in Squares) pulse(x[s], Size[s]);

cumulFunction ry = sum(s in Squares) pulse(y[s], Size[s]);

execute {

var f = cp.factory;

cp.setSearchPhases(f.searchPhase(x),

f.searchPhase(y));

};

constraints {

// for the cumulative function, from time 0 to 112, we must have the cumulative function always equal to 112

// what does this mean? it mean the resource is fully utilised at all time?

// each squre is scheduled between 0 and 112, and each square consume some space,

alwaysIn(rx,0,SizeSquare,SizeSquare,SizeSquare);

alwaysIn(ry,0,SizeSquare,SizeSquare,SizeSquare);

// what is ordered mean?

forall(ordered i, j in Squares)

// end of i must be fore j or end of j must be before i

// end of i in y must be before j in y or the

endOf(x[i]) <= startOf(x[j]) ||

endOf(x[j]) <= startOf(x[i]) ||

endOf(y[i]) <= startOf(y[j]) ||

endOf(y[j]) <= startOf(y[i]);

};