

Exercice 1 :

- 1-Commenter ce code, en précisant chaque partie correspond à quelle étape de la déclaration et/ou résolution du problème en utilisant Choco.
- 2-En déduire sa modélisation formelle. Reconnaissez-vous le problème xxxx?
- 3-Donner une solution de ce problème, en supposant que le solveur suit une méthode de filtrage par arc (ordre croissant des variables et valeurs).

```
7 public class exercice1 {
8
9     public void modelAndSolve(){
10         int n = 8;
11         Model model = new Model(n + "-xxxx problem");
12         IntVar[] vars = new IntVar[n];
13         for(int q = 0; q < n; q++){
14             vars[q] = model.intVar("Q_" + q, 1, n);
15         }
16         for(int i = 0; i < n-1; i++){
17             for(int j = i + 1; j < n; j++){
18                 model.arithm(vars[i], "!=", vars[j]).post();
19                 model.arithm(vars[i], "=", vars[j], "-", j - i).post();
20                 model.arithm(vars[i], "=", vars[j], "+", j - i).post();
21             }
22         }
23         Solution solution = model.getSolver().findSolution();
24         if(solution != null){
25             System.out.println(solution.toString());
26         }
27     }
28
29     public static void main(String[] args) {
30         new exercice1().modelAndSolve();
31     }
32
33 }
```

Exercice 2 :

Implémenter le problème d'ordonnancement suivant avec Choco :

Il y a 10 tâches:

A, B,C,D,E,F,G,H,I,J à exécuter de durées respectives : 5,4,3,2,1,5,4,3,2,1,

en respectant les contraintes de précédences :

A avant B, C, D -- B avant E -- C avant F, G -- D avant F -- E avant H -- F avant I -- G avant I -- H avant J -- I avant J.

Toutes les tâches doivent commencer dans un délai de 15 minutes.

Exercice 3 :

On note $w1(2)$ l'écriture par la transaction 1 d'un objet 2 et $r2(1)$ l'écriture par une transaction 2 d'un objet 1.

$w1(1), r2(1), w1(2), w3(3), r2(3), r4(2), w2(4), w4(5), r5(4), w5(5)$

- 1) Proposez un problème PSC qui cherche si une séquence d'opérations est sérialisable ou non. Implémenter sa solution en Choco.
- 2) Implémenter l'algorithme backtrack pour résoudre ce problème.
- 3) Donner la taille max dont on peut résoudre le problème en moins d'une seconde.

Pour mesure la temps d'exécution, on lancera au début du programme un chrono :

```
long start = System.currentTimeMillis();
```

Et à la fin du programme :

```
System.out.println("Temps d'exécution" + (System.currentTimeMillis() - start));
```