

Problème 1 :

On veut écrire un programme qui permette de renseigner les horaires d'ouverture d'un établissement. Les horaires sont stockés sous la forme suivante :

```
var openingHours = {  
  monday: [  
    { begin: "10:00", end: "12:00" },  
    { begin: "13:00", end: "18:00" },  
  ],  
  tuesday: [  
    { begin: "10:00", end: "12:00" },  
    { begin: "13:00", end: "18:00" },  
  ],  
  wednesday: [  
    { begin: "10:00", end: "12:00" },  
    { begin: "13:00", end: "18:00" },  
  ],  
  thursday: [  
    { begin: "10:00", end: "12:00" },  
    { begin: "13:00", end: "18:00" },  
  ],  
  friday: [],  
  saturday: [],  
  sunday: [],  
}
```

Ecrivez une méthode qui permette d'ajouter un horaire.

Cette méthode recevra deux objets en paramètre :

```
var daysSelected = {  
  monday: true,  
  tuesday: true,  
  wednesday: true,  
  thursday: true,  
  friday: false,  
  saturday: false,  
  sunday: false,  
}
```

```
var hoursToAdd = { begin: "19:00", end: "23:00" }
```

Amélioration 1

Cette méthode devra trier les horaires d'ouverture de chaque jour pour les afficher du plus tôt au plus tard.

Amélioration 2

Cette méthode devra vérifier que le nouvel horaire n'entre pas en collision avec les horaires déjà renseignés, si il y a au moins une collision, tout annuler et ne rien faire.

Exemple de collision : { begin: "10:00", end: "12:00" } n'est pas compatible avec { begin: "11:00", end: "13:00" }.

Problème 2 :

Nous avons un cube composé de petits cubes (ex : un rubiks cube).
Soit n le nombre de petits cubes dans une arête du grand cube.
(ex: pour un rubiks cube classique $n = 3$).

On veut calculer le nombre N de petits cubes à la périphérie du grand cube.
(ex pour $n = 2$, $N = 8$).

Ecrivez un programme qui permette de calculer ce nombre.

Problème 3 :

Ecrivez un programme qui permette d'imprimer la suite suivante :

```
1
2 3
4 5 6
7 8 9 10
...
```