Structures multidimensionnelles

Mardi 8 Janvier 2013

Option Informatique

Ecole Alsacienne

Bonne Année !

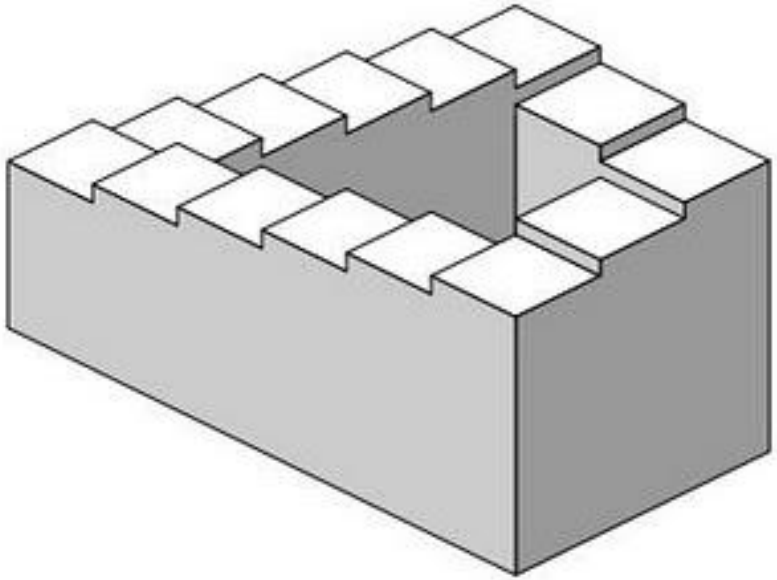
Plan

1. Des structures multidimensionnelles ?
2. Combiner vecteurs et listes
3. Les huit dames
4. Labyrinthes
5. Exercices

Des structures multidimensionnelles

POUR COMMENCER…

Qu'est-ce que c'est que cette histoire de multidimension ?



Dimension 1

Une **structure à une dimension**, c'est tout simplement :

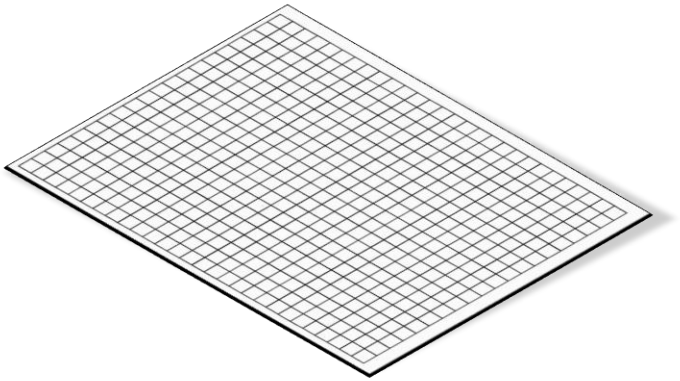
- Une droite
- Un vecteur
- Une liste
- Informellement, un truc "rectiligne"

0	1	2	3	4	5	6	7	8	9
51	23	78	64	89	53	90	12	3	75

DIMENSION 2

Une **structure à deux dimension**, c'est par exemple :

- Un plan
- Un tableau à double entrée

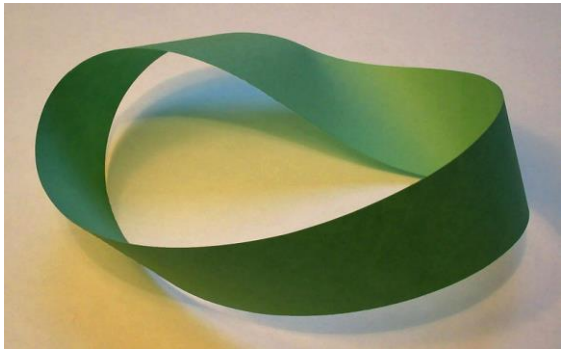


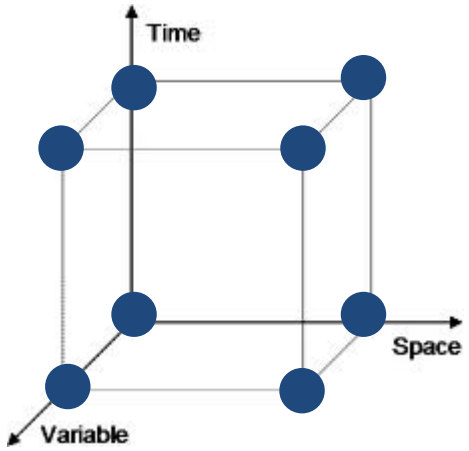
	USA	Chine	Russie
Prénom	Barack	Hu	Vladimir
Nom	Obama	Jintao	Poutine

DIMENSION 3

Une **structure à trois dimensions**, c'est par exemple :

- Un objet en relief
  - N'importe quel objet en relief
  - Enfin presque…
- Un tableau à 3 entrées

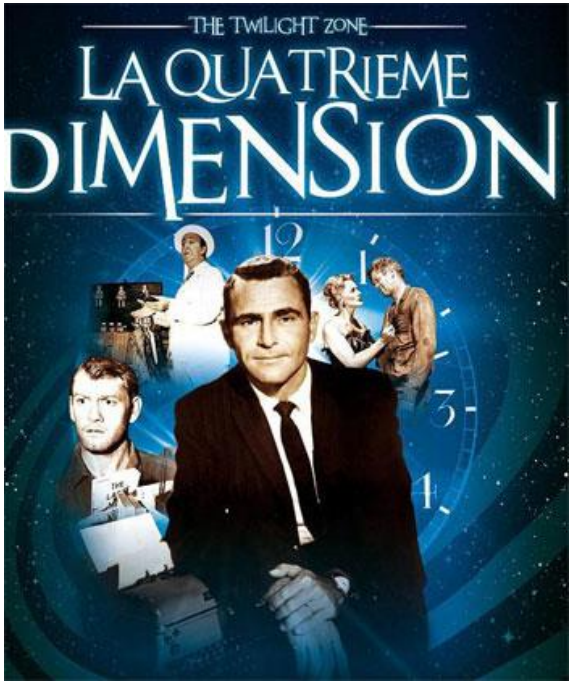




DIMENSION 4 (ET PLUS)

Au-delà de la dimension 3, **les choses se compliquent**…

- La représentation mentale et physique est ardue
- Mais on pourra créer et manipuler de tels objets virtuels



DIMENSION 0

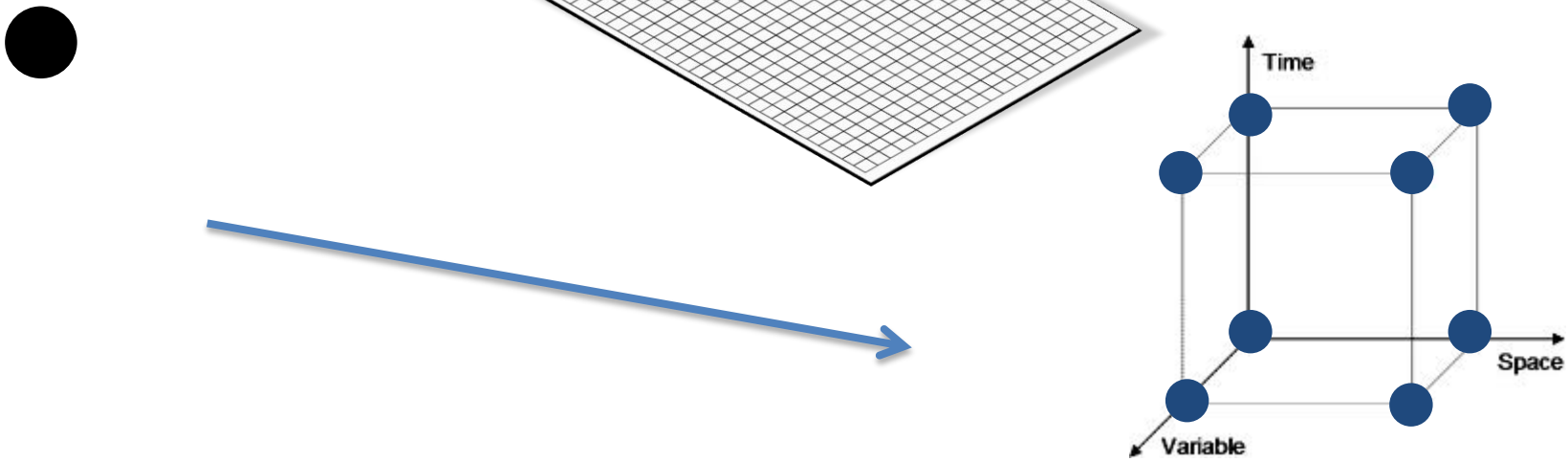
Une structure de dimension 0, c'est :

- Un point
- Une variable simple (ex : `int`, `float`, `bool`)

FORMELLEMENT

La **dimension** d'un objet mathématique est

- Le **nombre de degrés de liberté** qu'on a avec cet objet
- Le **nombre de paramètres à fixer pour se situer** sur cet objet



Combiner vecteurs et listes

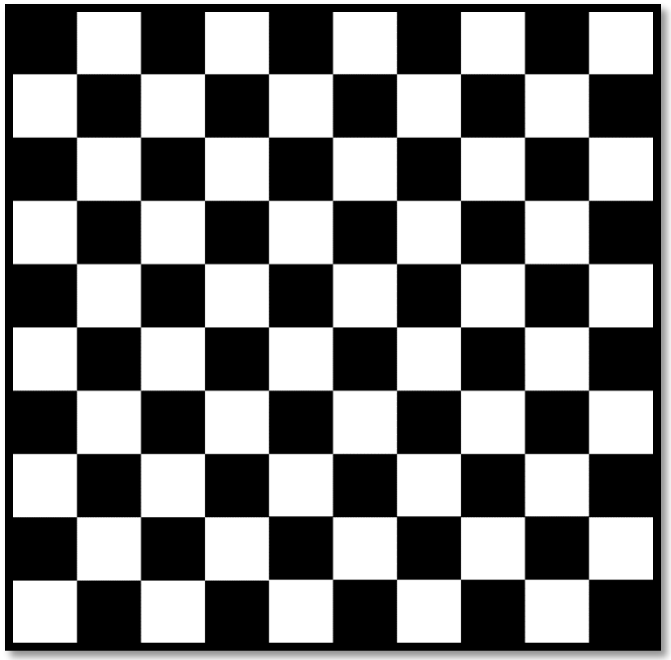
Comment passer en dimensions supérieures ?

- *Question : Comment utiliser les structures de données que nous avons déjà vues (vecteurs et listes) pour obtenir des structures à plusieurs dimensions ?*

- Réponse : en les **combinant** !
  - Vecteur de vecteurs
  - Vecteur de listes
  - Listes de vecteurs
  - Listes de listes
  - Etc.

Exemple 1 : le damier

- *Question : Quelle structure vous semblerait adaptée pour représenter l'objet suivant ?*



Vecteur de vecteurs

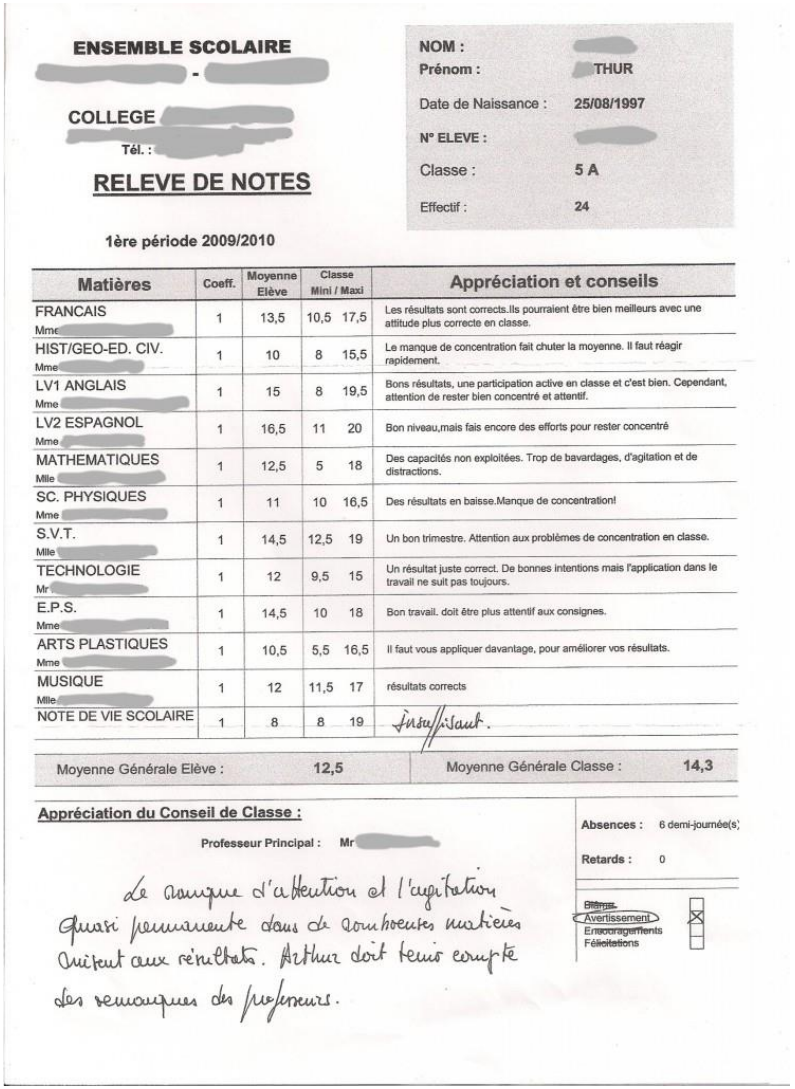
EXEMPLE 2 : LE RÉPERTOIRE ALPHABÉTIQUE

- *Question : Quelle structure vous semblerait adaptée pour représenter l'objet suivant ?*

Vecteur de listes

Exemple 3 : Le relevé de notes

- *Question : Quelle structure vous semblerait adaptée pour représenter l'objet suivant ?*



ENSEMBLE SCOLAIRE [redacted]

COLLEGE [redacted]
Tél. : [redacted]

RELEVE DE NOTES

NOM : [redacted]
Prénom : [redacted]THUR
Date de Naissance : 25/08/1997
N° ELEVE : [redacted]
Classe : 5 A
Effectif : 24

1ère période 2009/2010

Matières	Coeff.	Moyenne Elève	Classe Mini / Maxi	Appréciation et conseils
FRANCAIS Mme[redacted]	1	13,5	10,5 17,5	Les résultats sont corrects.Ils pourraient être bien meilleurs avec une attitude plus correcte en classe.
HIST/GEO-ED. CIV. Mme[redacted]	1	10	8 15,5	Le manque de concentration fait chuter la moyenne. Il faut réagir rapidement.
LV1 ANGLAIS Mme[redacted]	1	15	8 19,5	Bons résultats, une participation active en classe et c'est bien. Cependant, attention de rester bien concentré et attentif.
LV2 ESPAGNOL Mme[redacted]	1	16,5	11 20	Bon niveau,mais fais encore des efforts pour rester concentré
MATHEMATIQUES Mlle[redacted]	1	12,5	5 18	Des capacités non exploitées. Trop de bavardages, d'agitation et de distractions.
SC. PHYSIQUES Mme[redacted]	1	11	10 16,5	Des résultats en baisse.Manque de concentration!
S.V.T. Mlle[redacted]	1	14,5	12,5 19	Un bon trimestre. Attention aux problèmes de concentration en classe.
TECHNOLOGIE Mr[redacted]	1	12	9,5 15	Un résultat juste correct. De bonnes intentions mais l'application dans le travail ne suit pas toujours.
E.P.S. Mme[redacted]	1	14,5	10 18	Bon travail. doit être plus attentif aux consignes.
ARTS PLASTIQUES Mme[redacted]	1	10,5	5,5 16,5	Il faut vous appliquer davantage, pour améliorer vos résultats.
MUSIQUE Mlle[redacted]	1	12	11,5 17	résultats corrects
NOTE DE VIE SCOLAIRE	1	8	8 19	Insuffisant.

Moyenne Générale Elève : 12,5 — Moyenne Générale Classe : 14,3

Appréciation du Conseil de Classe :

Professeur Principal : Mr[redacted]

Le manque d'attention et l'agitation quasi permanente dans de nombreuses matières nuisent aux résultats. Arthur doit tenir compte des remarques des professeurs.

Absences : 6 demi-journée(s)
Retards : 0

Blâme
Avertissement
Encouragements
Félicitations

Liste de vecteurs

Exemple 4 : les évènements et les participants

- *Question : Quelle structure vous semblerait adaptée pour représenter l'objet suivant ?*
  - *Plusieurs ateliers*
  - *Plusieurs participants*

Listes de listes

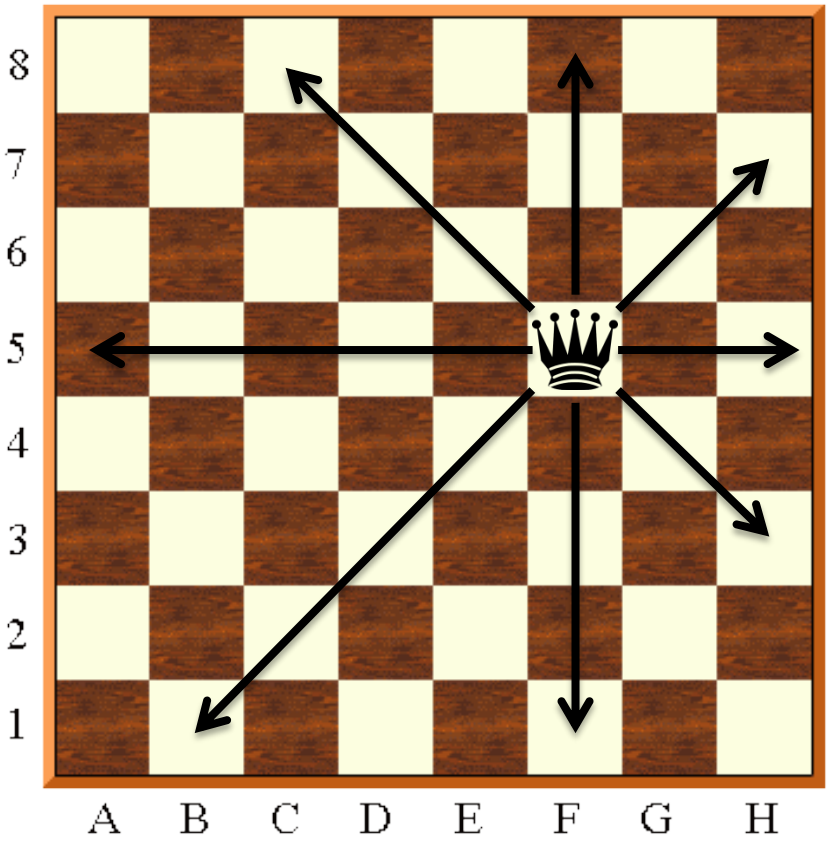
Que privilégier ?

- Structure de taille fixe (connue à l'avance) ?
  - ➔ Vecteurs

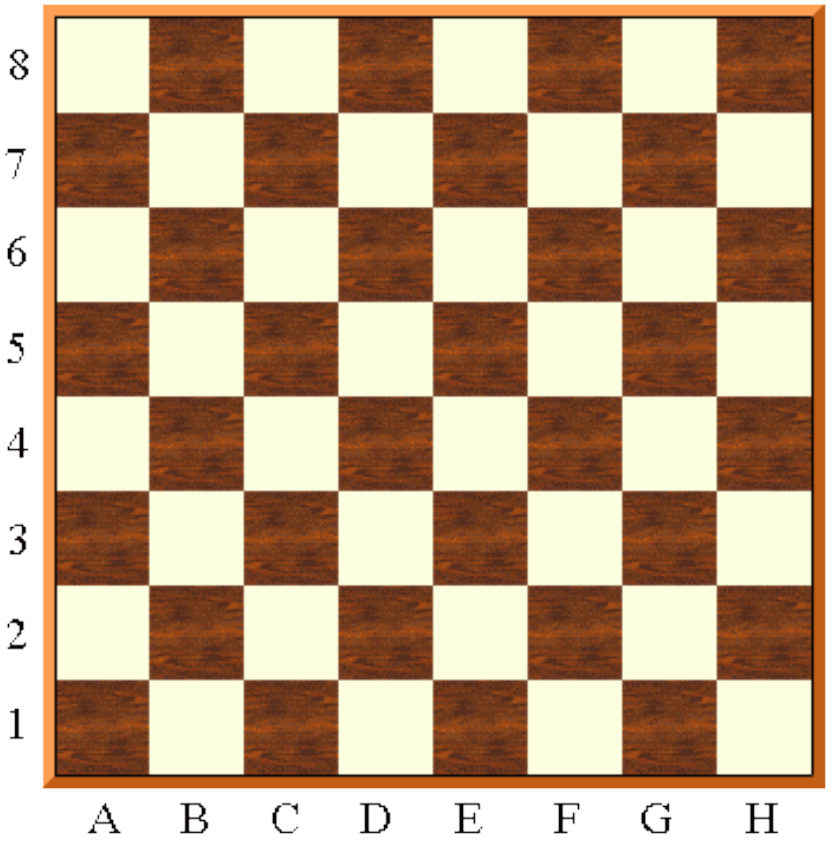
- Nombre d'éléments inconnus à l'avance
  - ➔ Listes

Les huit dames

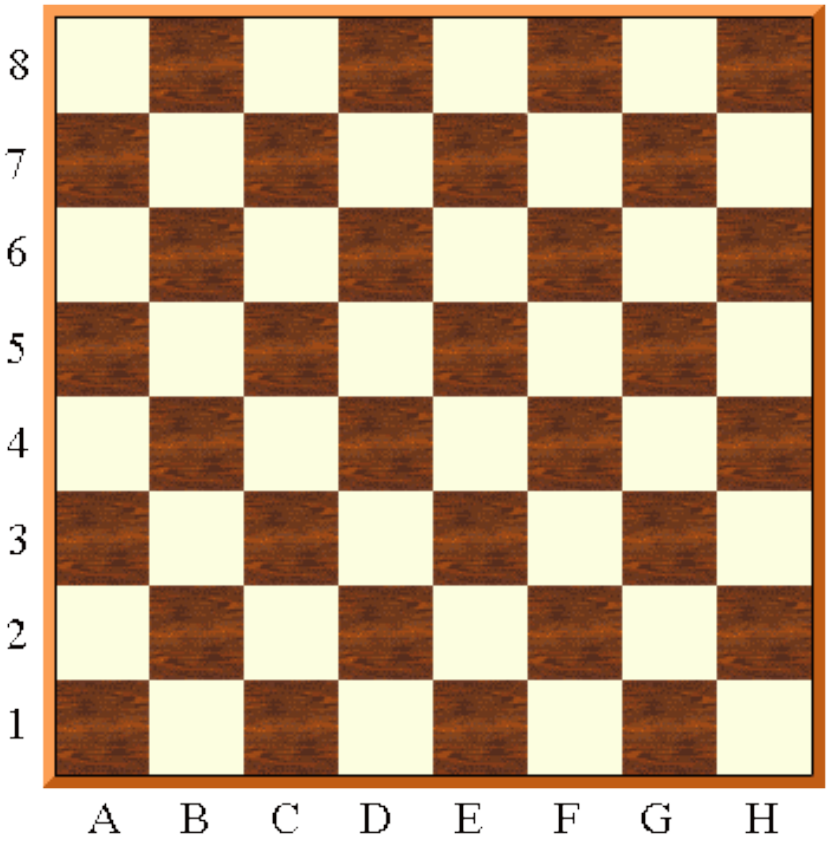
La dame aux échecs



Avec deux dames

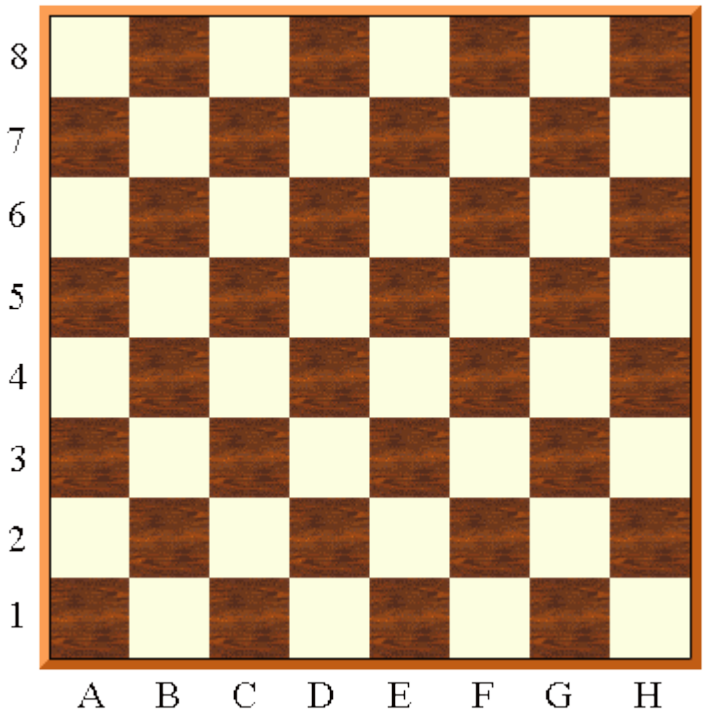


AVEC HUIT DAMES



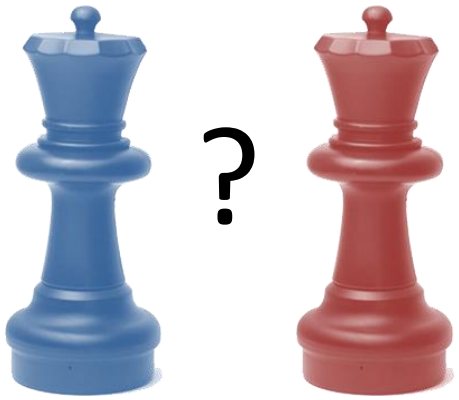
Le problème des huit dames

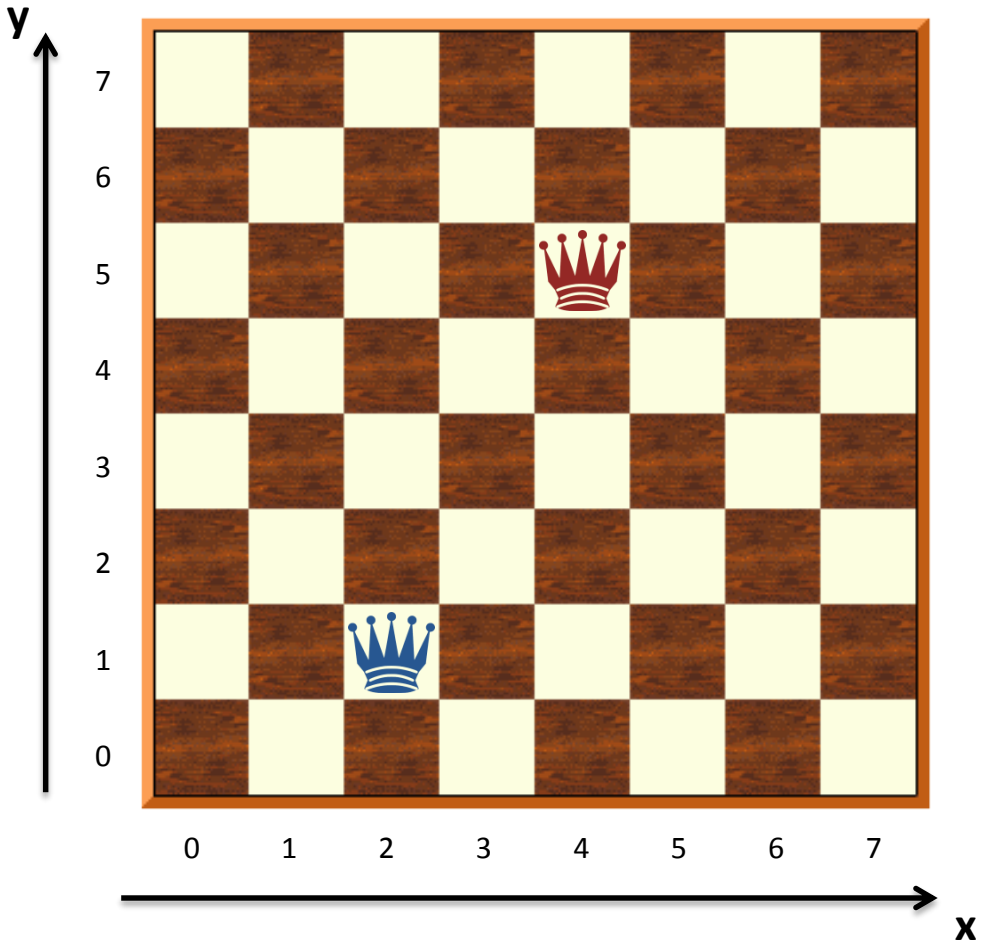
- *Question : Combien de façons existe-t-il de poser huit dames sur un échiquier sans qu'aucune ne soit en prise avec une autre ?*



COMMENT SAVOIR SI DEUX DAMES SONT EN PRISE ?

- *Question : Comment déterminer si les dames situées aux positions* (x_1, y_1) *et* (x_2, y_2) *sont en prise ?*





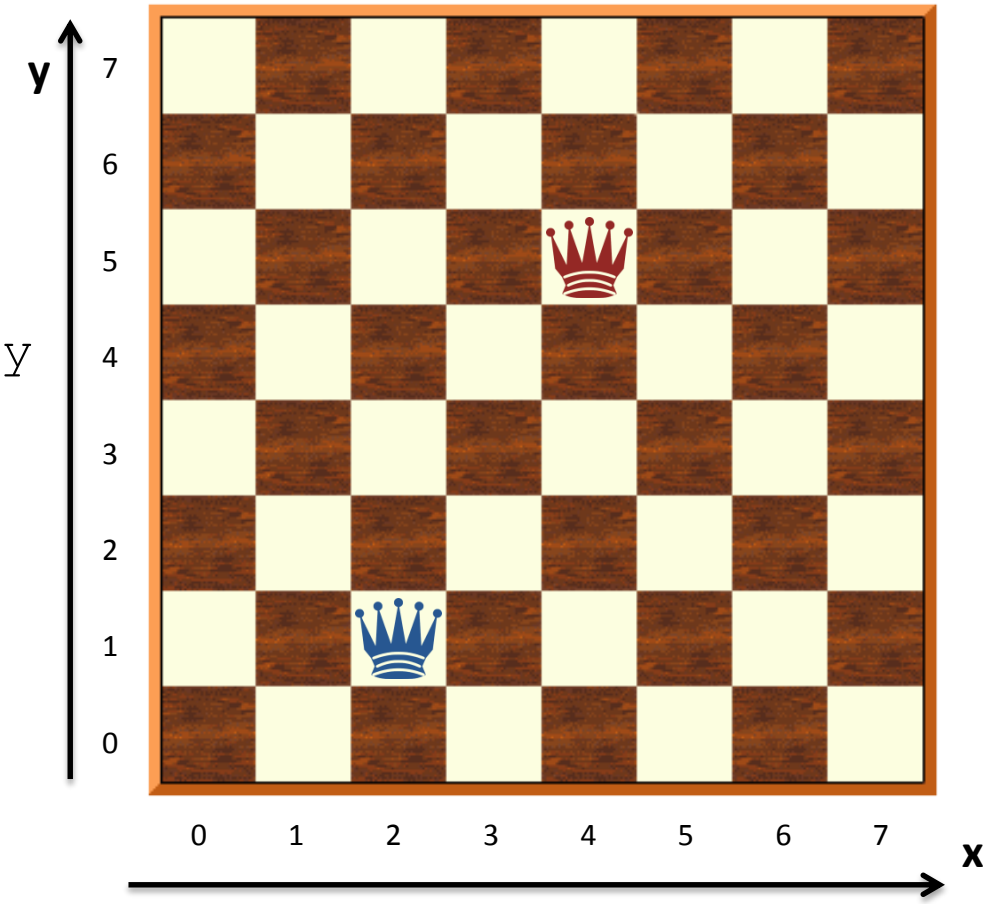
Représentation du damier

- L'échiquier est représenté par un vecteur de vecteurs

- Pour chaque case, on utilise un booléen pour indiquer la présence d'une dame :
  - Vrai si une dame est posée
  - Faux sinon

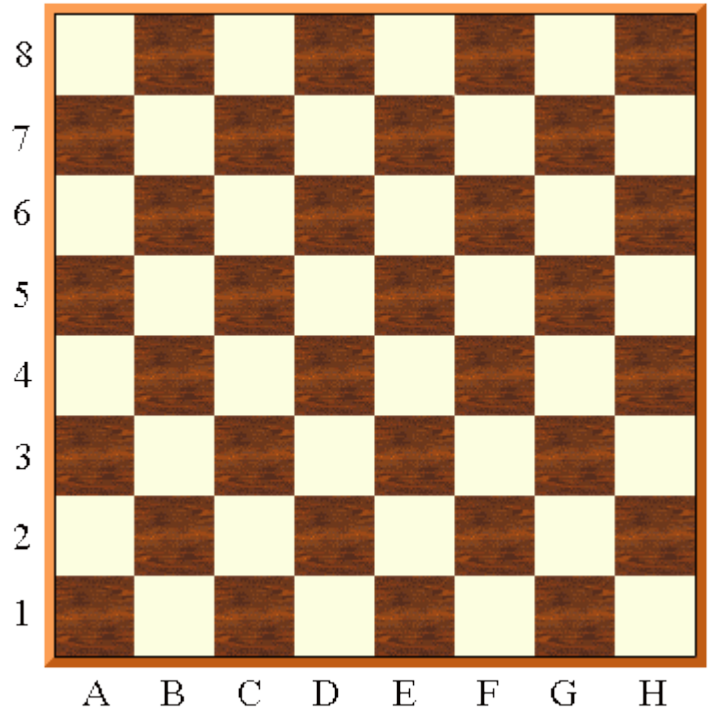
- En Caml, `bool array array`

- Exemples
  - `damier.(2).(2) = false`
  - `damier.(4).(5) = true`



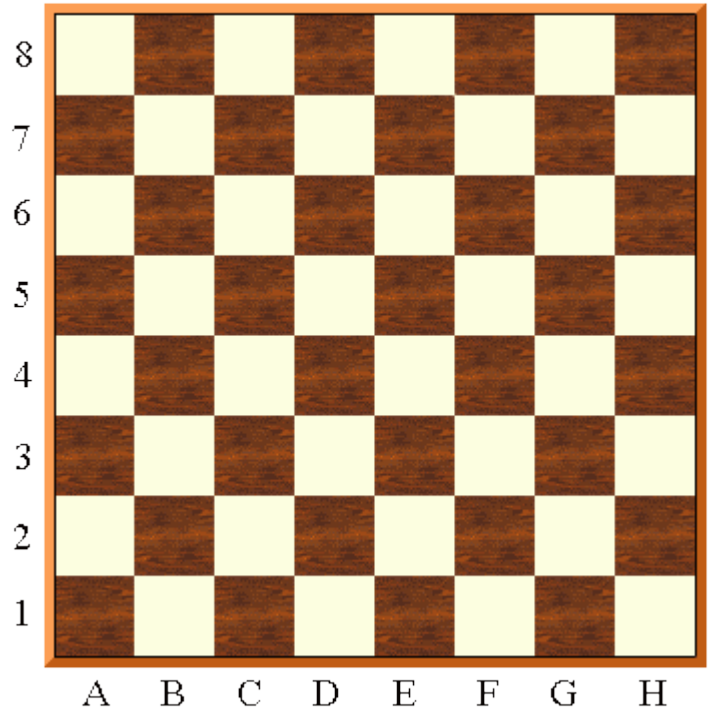
PARLONS CHIFFRES…

- Approche naïve : *Combien y a-t-il de façons de poser ces huit dames sur l'échiquier ?*
  - ➔ $64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57 = 178\,462\,987\,637\,760$
  - ➔ *S'il faut une nanoseconde (10^{-9} sec) pour tester chaque disposition, il faudrait plus de deux jours avec cette méthode*



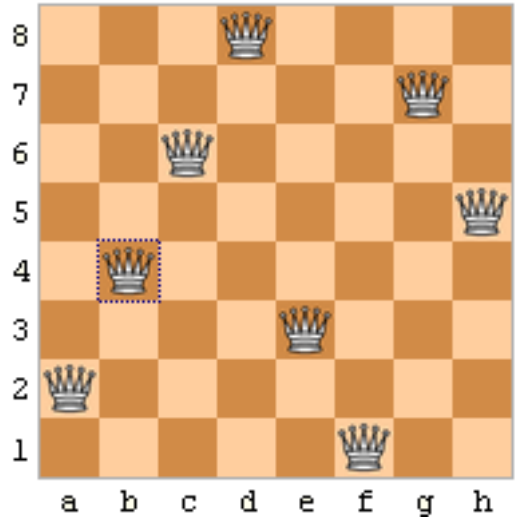
PARLONS CHIFFRES…

- Approche plus efficace : *Combien y a-t-il de façons de poser ces huit dames sur l'échiquier, sans considération d'ordre ?*
  - ➔ $\binom{64}{8} = \frac{64!}{8! \times (64-8)!} = \frac{64 \times 63 \times 62 \times 61 \times 60 \times 59 \times 58 \times 57}{8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1} = 4\,426\,165\,368$
  - ➔ *S'il faut une nanoseconde (10^{-9} sec) pour tester chaque disposition, il faudra quelques secondes avec cette méthode*



Résultats pour un échiquier classique

- Nombre de solutions pour un échiquier classique (8 cases) :
  - ➔ 92 solutions



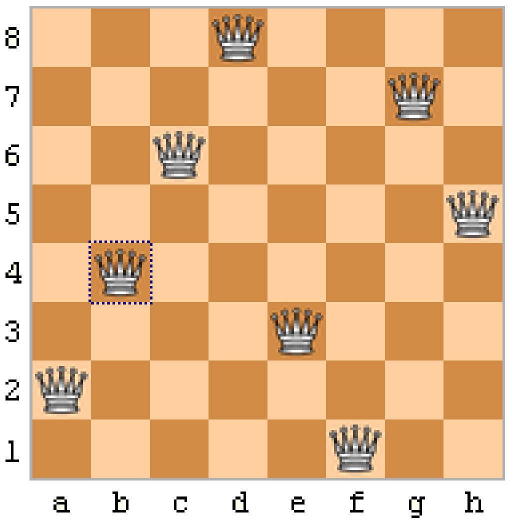
- Arguments de symétries :
  - Symétrie centrale : avec une rotation de 90°, on obtient une autre solution
  - Symétrie axiale : en inversant haut et bas ou gauche et droite, on obtient une autre solution

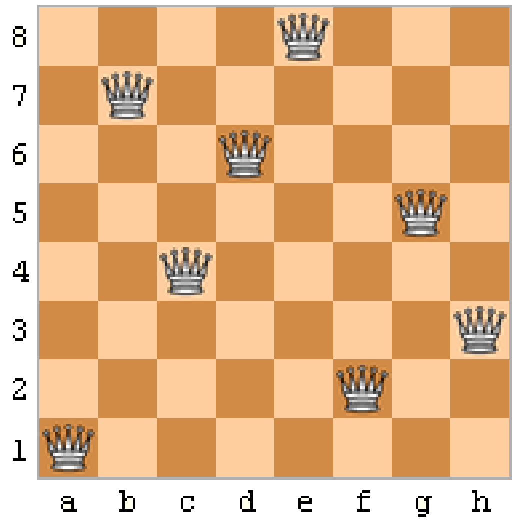
- Nombre de solutions « aux symétries » près :
  - ➔ 12 solutions

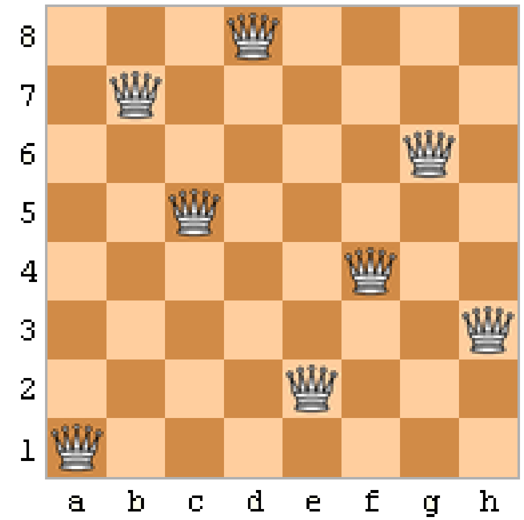
- Génération du problème :

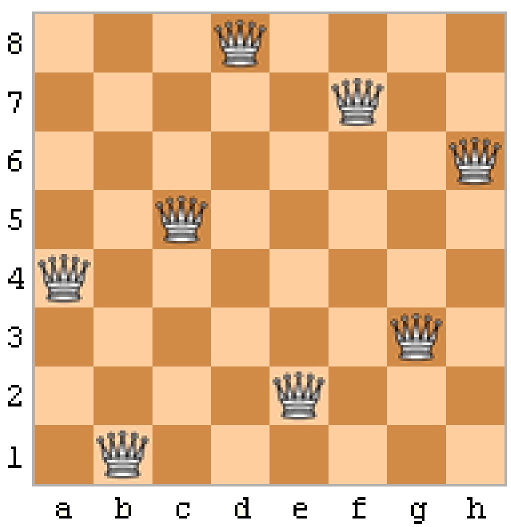
Combien de façons existe-t-il de poser huit dames sur un échiquier **à n cases** *sans qu'aucune ne soit en prise avec une autre ?*

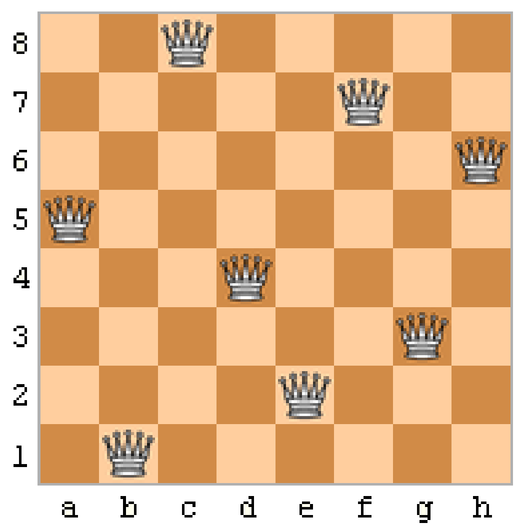
Les 12 solutions

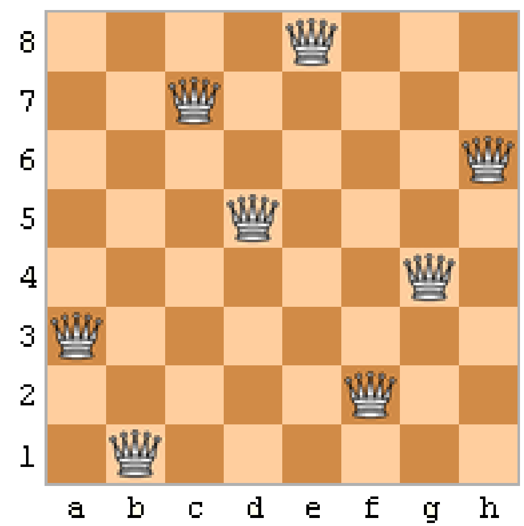




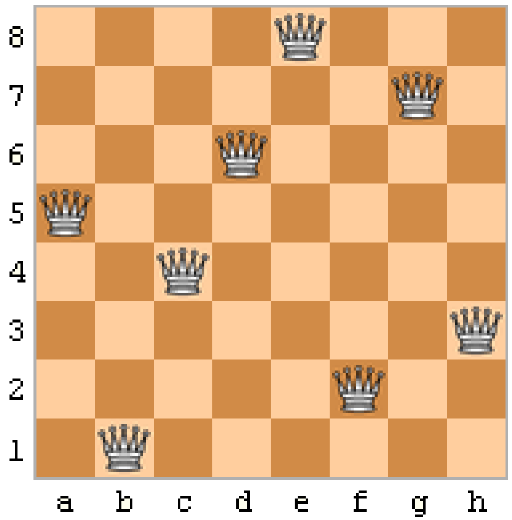


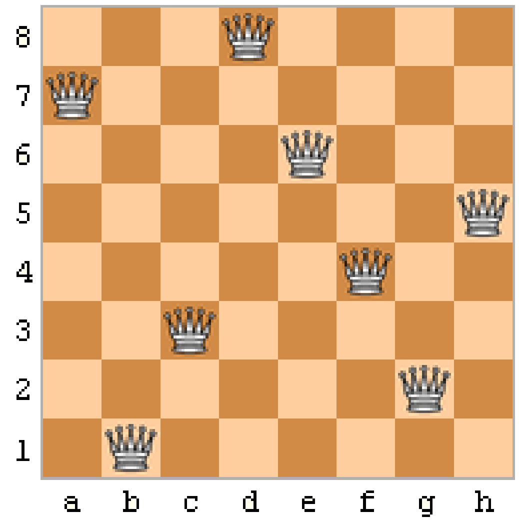


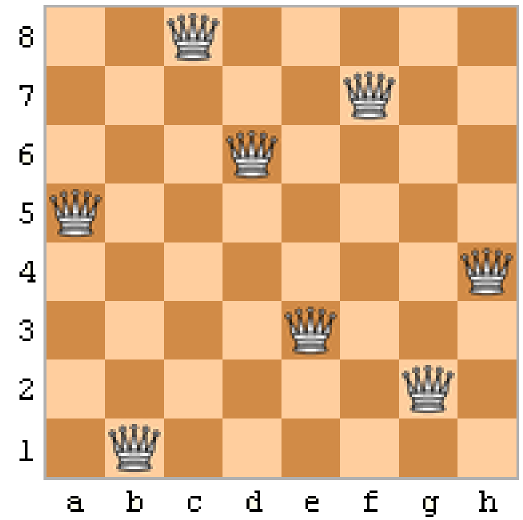


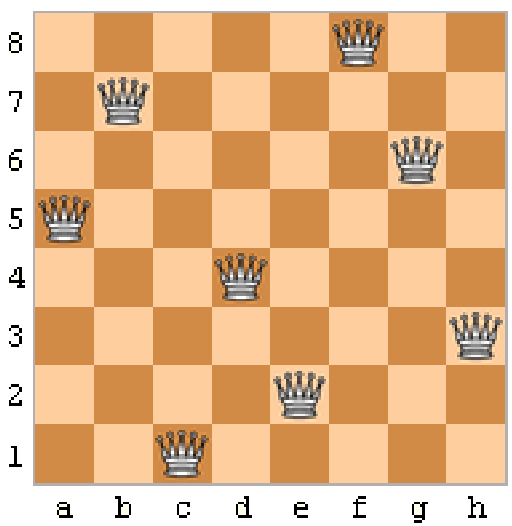


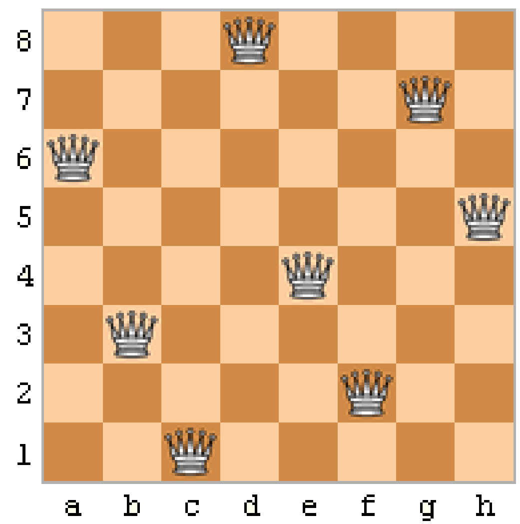
Les 12 solutions

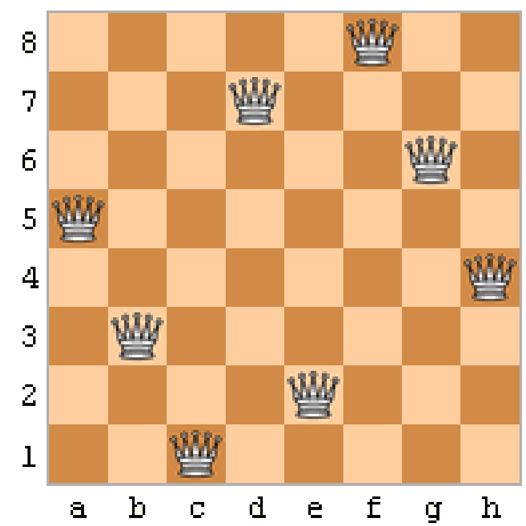






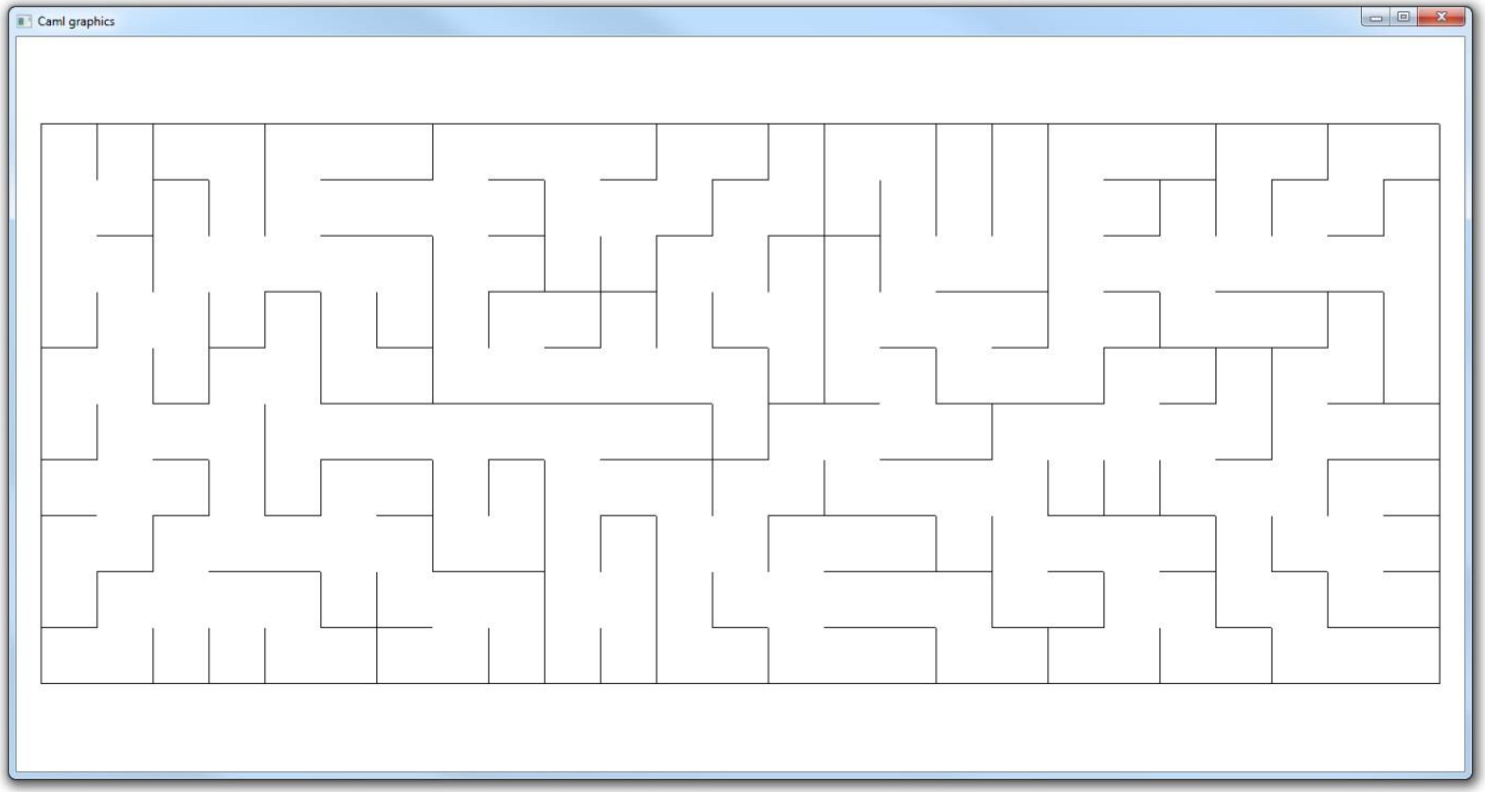






Labyrinthes

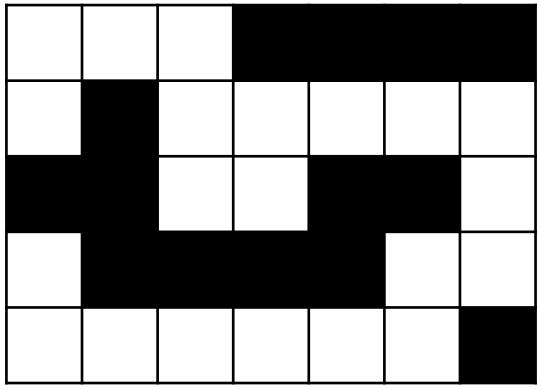
NOTRE OBJECTIF

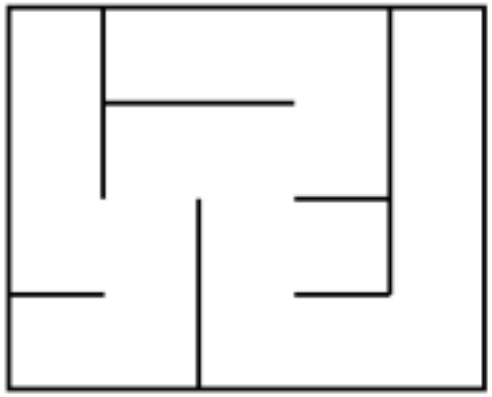


Choix de la représentation

- **Question** : *Quelle structure de données pourrait-on utiliser pour représenter un labyrinthe ?*
  - ➔ **Réponse** : un vecteur de vecteurs

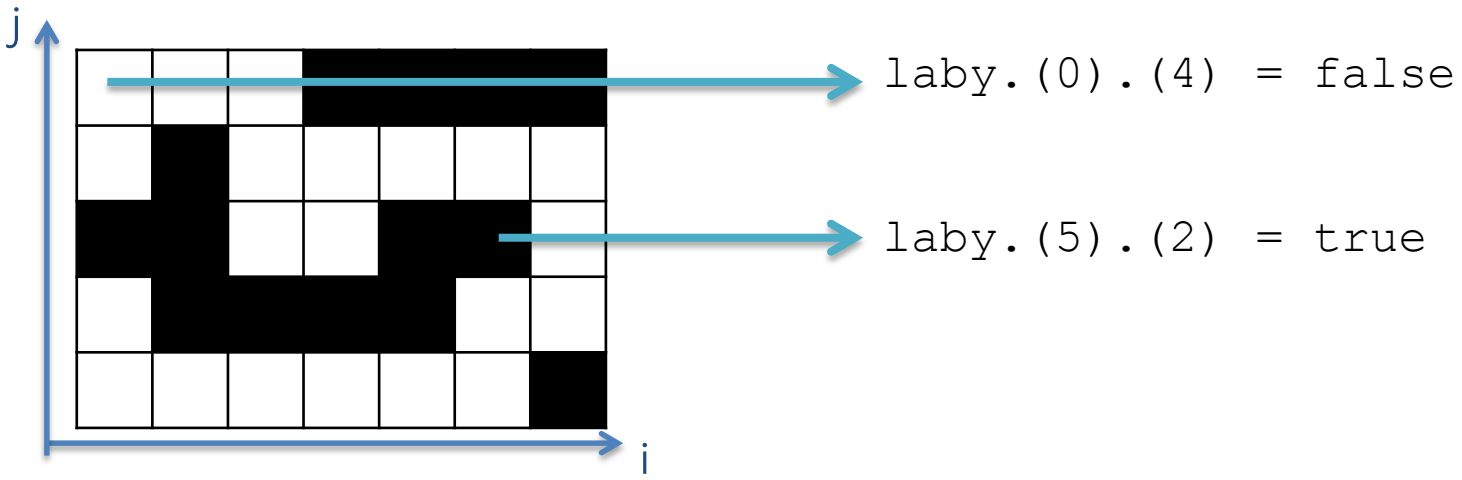
- **Question subsidiaire** : *Comment représenter chaque case ?*
  - ➔ **Première réponse possible** : avec un booléen
  - ➔ **Deuxième réponse possible** : avec plusieurs booléen





LABYRINTHES SIMPLES : AVEC UN SEUL BOOLÉEN

- **Question** : *Comment représenter un labyrinthe avec un seul booléen pour chaque case ?*



- Convention proposée :

  `laby.(i).(j) = true` si la case contient un mur, `false` sinon

- Type correspondant en Caml : `bool array array`

Labyrinthes simples : Problème de voisinage

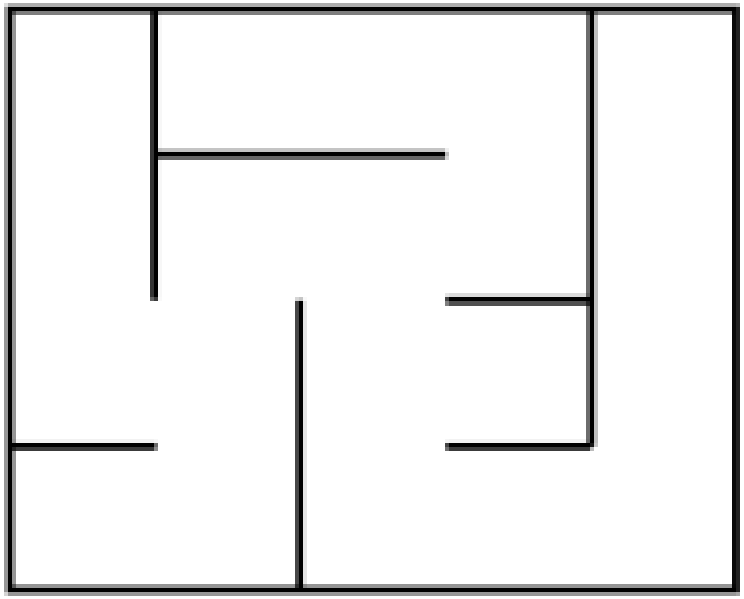
- Question : *Comment déterminer les voisins accessibles à partir d'une case donnée ?*

- Réponse :

```
let voisins laby i j =

  let l = largeur laby in
  let h = hauteur laby in
  let v = ref [] in

  (* Case de gauche *)
  if ( (i>0) && (not laby.(i-1).(j)) )
  then v := (i-1,j)::!v;

  (* Case de droite *)
  if ( (i<l-1) && (not laby.(i+1).(j)) )
  then v := (i+1,j)::!v;

  (* Case du bas *)
  if ( (j>0) && (not laby.(i).(j-1)) )
  then v := (i,j-1)::!v;

  (* Case du haut *)
  if ( (j<h-1) && (not laby.(i).(j+1)) )
  then v := (i,j+1)::!v;

  !v;;
```

Labyrinthe avancés

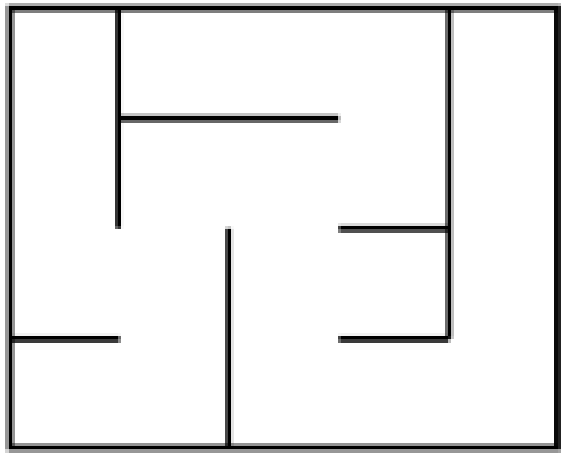
- Principe : *Les murs sont situés entre les cases*



Labyrinthe avancés : Choix de la représentation

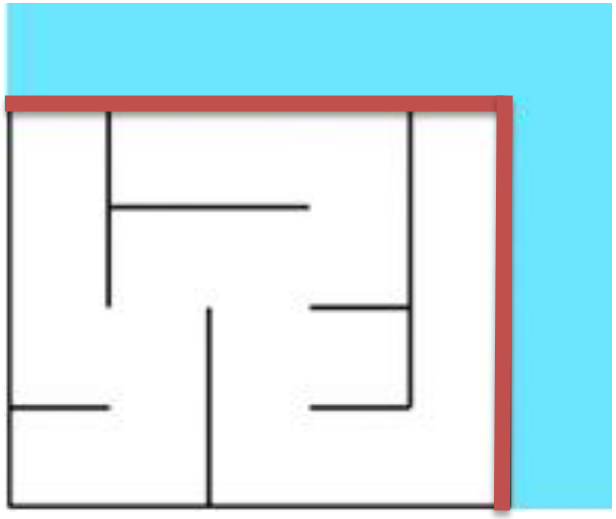
- Question : *Comment représenter un tel labyrinthe ?*

- Réponse :
  - On utilise un vecteur de vecteurs
  - Pour chaque case, on stocke deux informations :
    - *Est-ce qu'il y a un mur à gauche de cette case ?*
    - *Est-ce qu'il y a un mur au-dessous de cette case ?*



Labyrinthe avancés : Choix de la représentation

- Question : *Est-ce qu'on va pouvoir gérer toutes les cases avec cette représentation ?*



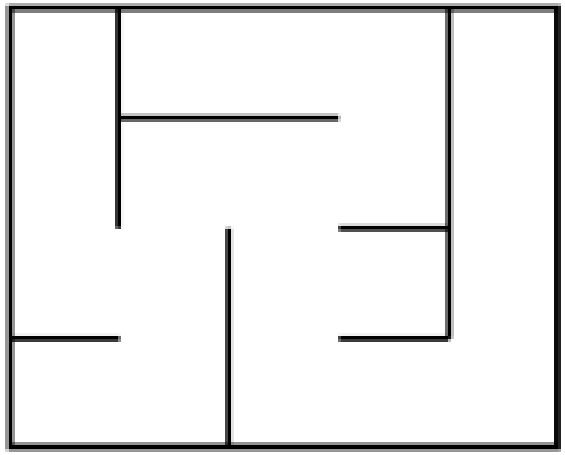
- Réponse : *Oui, en ajoutant une ligne et une colonne*
  - *Des cases qui servent juste à fermer le labyrinthe*
  - *Des cases auxquelles on n'accèdera pas par la suite*

Labyrinthe avancés : Implémentation en Caml

- **Une idée clef en programmation** :

Créer des objets et des types adaptés à nos besoins.

- Pour nos labyrinthes, on va créer le type `case`, avec deux propriétés : `gauche` et `bas` (de type `bool`)



Labyrinthe avancés : Implémentation en Caml

- Savoir si un mur existe (renvoie un booléen)
  - *Mur gauche d'une case :* `laby.(i).(j).gauche`
    - Exemple : `if (laby.(1).(3).gauche)`
  - *Mur bas d'une case :* `laby.(i).(j).bas`
    - Exemple : `if (not laby.(3).(7).bas)`
- Ajouter ou supprimer un mur :
  - *Mur gauche d'une case :* `laby.(i).(j).gauche <- valeur;`
    - Exemple : `laby.(1).(3).gauche <- true;`
  - *Mur bas d'une case :* `laby.(i).(j).bas <- valeur;`
    - Exemple : `laby.(1).(3).gauche <- true;`

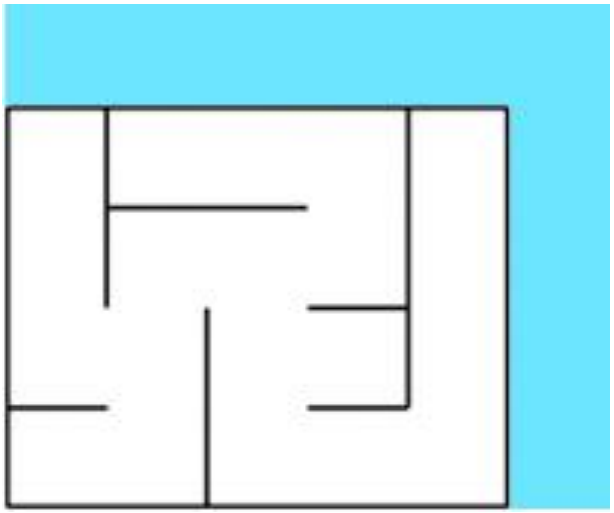
Labyrinthe avancés : Premières fonctions

- Comment tester s'il y a un mur à droite d'une case ?

```
let mur_a_droite laby i j =
    laby.(i+1).(j).gauche;;
```

- Comment tester s'il y a un mur au-dessus d'une case ?

```
let mur_au_dessus laby i j =
    laby.(i).(j+1).bas;;
```

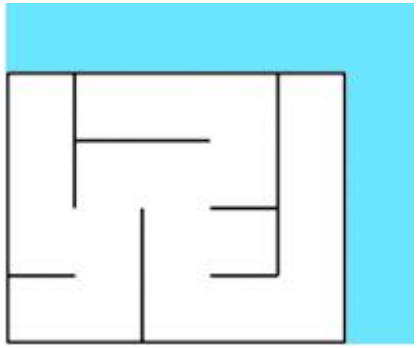


LABYRINTHE AVANCÉS : PROBLÈME DE VOISINAGE

- Question : *Comment déterminer les voisins accessibles à partir d'une case donnée ?*

- Réponse :

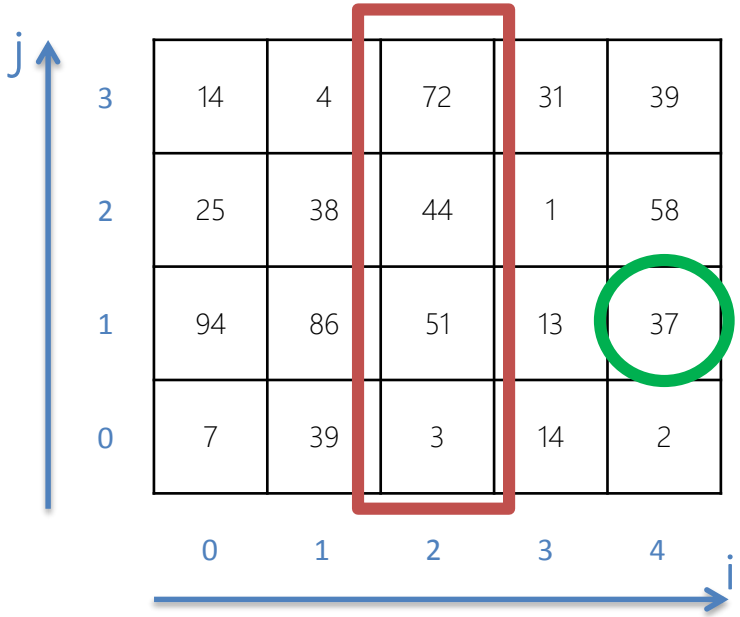
```
let voisins laby i j =

  let l = largeur laby in
  let h = hauteur laby in
  let v = ref [] in

  (* Case de gauche *)
  if ( (i>0) && (not laby.(i).(j).gauche) )
  then v := (i-1,j)::!v;

  (* Case de droite *)
  if ( (i<l-1) && (not (mur_a_droite laby i j)) )
  then v := (i+1,j)::!v;

  (* Case du bas *)
  if ( (j>0) && (not laby.(i).(j).bas) )
  then v := (i,j-1)::!v;

  (* Case du haut *)
  if ( (j<h-1) && (not (mur_au_dessus laby i j)) )
  then v := (i,j+1)::!v;

  !v;;
```



Exercices

Un vecteur de vecteurs ?

- Un tableau à deux entrées est un vecteur de vecteurs :

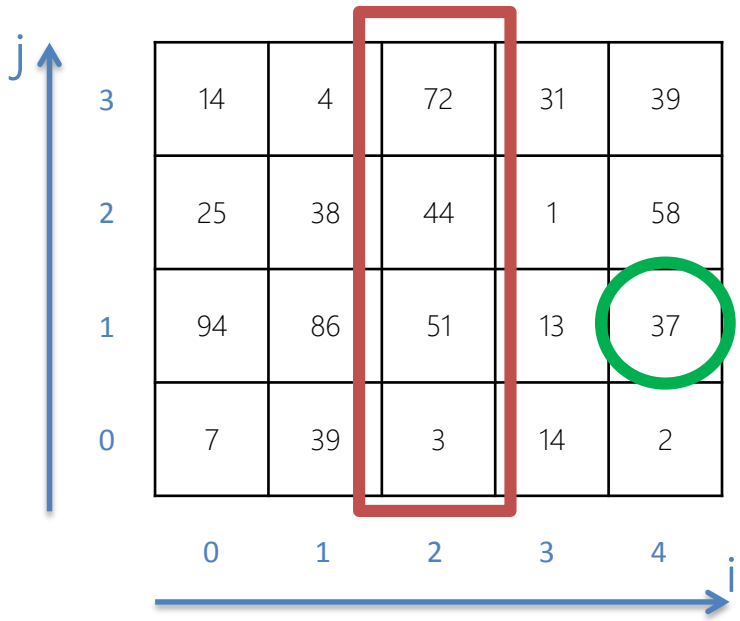


- Si `t` représente tout le tableau (`int array array`)
  - `t.(4).(1)` représente une case contenant 37 (`int`)
  - `t.(2)` représente la troisième colonne du tableau (`int array`)
  - `t` est en quelque sort un « vecteur de colonnes »

Largeur et hauteur d'un tableau

- **Question** : *Comment trouver la largeur et la hauteur d'un tableau (représenté par un vecteur de vecteurs) ?*

```
let largeur t =
    Array.length t;;

let hauteur t =
    Array.length t.(0);;
```



MAXIMUM D'UN VECTEUR DE VECTEURS

- Question : *Comment trouver la valeur maximale dans un tableau ?*

```
let maximum_tableau t =

  let l = largeur t in
  let h = hauteur t in
  let maximum = ref t.(0).(0) in

  for i = 0 to longueur1 - 1
  do
    for j = 0 to longueur2 - 1
    do
      if (t.(i).(j) > !maximum)
      then maximum := t.(i).(j);
    done
  done;

  !maximum;;
```

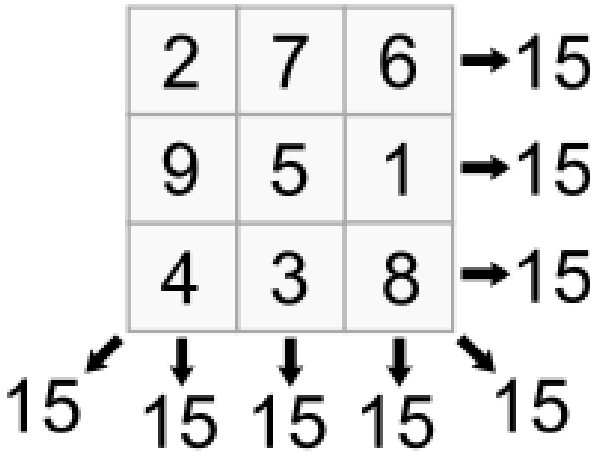
MAXIMUM D'UNE LISTE DE LISTES D'ENTIERS

- Question : *Comment trouver la valeur maximale dans une liste de listes de nombres entiers ?*

```
let rec maximum_liste l =
  if (l = [])
  then
    min_int
  else
    max (List.hd l)
        (maximum_liste (List.tl l));;

let rec maximum_liste_de_listes l =
  if (l = [])
  then
    min_int
  else
    max (maximum_liste (List.hd l))
        (maximum_liste_de_listes (List.tl l));;
```

Carrés magiques

- **Définition :** Un carré magique de taille n est une grille carrée de taille n contenant des entiers telle que les sommes de chaque ligne, de chaque colonne, et de chaque diagonale ont toutes la même valeur



25	13	1	19	7
16	9	22	15	3
12	5	18	6	24
8	21	14	2	20
4	17	10	23	11

CARRÉS MAGIQUES

- Question : *Comment tester si un vecteur de vecteur représente un carré magique ?*

```
let est_carre_magique t =
  let n = Array.length t in
  if (n <> Array.length t.(0))
  then false
  else
    begin
      let carre_magique = ref true in
      let total = total_colonne t 0 n in

      (* Test sur les colonnes *)
      for i = 1 to (n-1)
      do
       if (total <> (total_colonne t i n))
       then carre_magique := false;
      done;

      (* Test sur les lignes *)
      for j = 0 to (n-1)
      do
       if (total <> (total_ligne t j n))
       then carre_magique := false;
      done;

      (* Test sur les diagonales *)
      if (total <> (total_diag1 t n))
      then carre_magique := false;
      if (total <> (total_diag2 t n))
      then carre_magique := false;

      !carre_magique
    end;;
```

CARRÉS MAGIQUES

```
let total_colonne t i n =
  let total = ref 0 in
  for j = 0 to (n-1)
  do
    total := !total + t.(i).(j)
  done;
  !total;;

let total_ligne t j n =
  let total = ref 0 in
  for i = 0 to (n-1)
  do
    total := !total + t.(i).(j)
  done;
  !total;;

let total_diag1 t n =
  let total = ref 0 in
  for k = 0 to (n-1)
  do
    total := !total + t.(k).(k)
  done;
  !total;;

let total_diag2 t n =
  let total = ref 0 in
  for k = 0 to (n-1)
  do
    total := !total + t.(k).(n-1-k)
  done;
  !total;;
```

Prochaine séance

Mardi 15 Janvier 2013

[TD] Dames et labyrinthes