

Algorithmes et structures de données : TD 3

Types - Tableaux 1D - Tableaux 2D

Exercice 3.1 *Types*

Déclarer des types qui permettent de stocker :

1. Les états de la matière : liquide, solide, gaz.
2. Les 5 dés d'un jeu de yahtzee.
3. Une image noir et blanc de 512x512 pixels.
4. Un damier d'échec.

Exercice 3.2 *Tableau 1D*

Considérez le tableau 1D suivant :

```
type t_champ = (vide,vert,rouge);  
var tableau : array[1..6] of t_champ;
```

1. Ecrire un algorithme qui compte le nombre d'entrées de couleur verte dans le tableau.
2. Ecrire un algorithme qui affiche à l'écran la couleur qui apparaît plus souvent dans le tableau ou "égalité" sinon. Faites tourner votre algorithme dans un tableau.
3. Ecrire un algorithme qui parcourt le **tableau** et qui identifie la longueur maximale d'une suite d'entrées de **champ rouge** consécutive. Utilisez trois variables **i**, **longueur**, **max_longueur**. Faites tourner votre algorithme pour le tableau (rouge, vert, rouge, rouge, vert, rouge).
4. Ebaucher l'occupation de la mémoire d'un ordinateur avec 1 Mo de mémoire vive.

Exercice 3.3 *Tableau 2D*

Considérez le damier d'un jeu de dame :



1. Définissez les types pour stocker le damier.
2. Ebaucher l'occupation de la mémoire d'un ordinateur avec 1 Mo de mémoire vive.
3. Ecrire un algorithme qui initialise le damier de dame en posant les pions noirs et blanc sur le damier conforme à la Figure.
4. Ecrire un algorithme qui répond à l'écran à la question "qui est en train de gagner", c'est-à-dire de quelle couleur il y a plus de pions sur le damier.