

# INFO-H-100

## Séances d'exercices 8

Les tableaux (2) :

*Les matrices*

# Matrices

- Une matrice est un tableau à **deux dimensions** (un vecteur de vecteurs)

```
int M[5][10];
```

- Même principe que les vecteurs, sauf qu'une case est définie par deux indices : **ligne** et **colonne**.

```
M[4][9] = 8;
```





# Parcours d'une matrice (1)

```
int M[5][10];  
  
for(int i=0; i < 5; i++)  
{  
    for(int j=0; j < 10; j++)  
    {  
        M[i][j]=0;  
    }  
}
```











# Représentation mémoire

- Une matrice est une **abstraction** pour le **programmeur**.

<b>M</b>	0	1	2
0	a	b	c
1	d	e	f

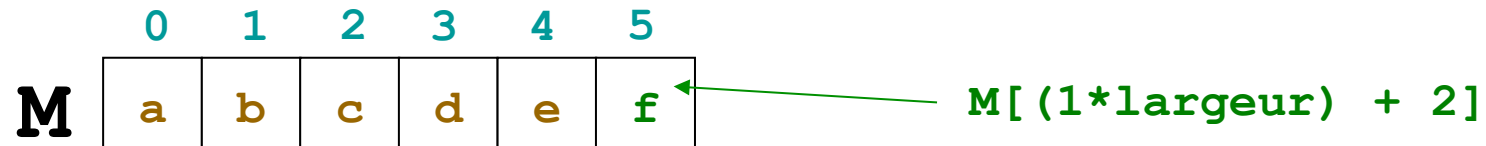
$M[1][2]$

- En **mémoire**, une matrice est représentée comme un vecteur, de **façon contiguë**.

	0	1	2	3	4	5
<b>M</b>	a	b	c	d	e	f

$M[(1 * \text{largeur}) + 2]$

# Matrice et fonctions



- Du fait de la représentation mémoire contiguë, le compilateur doit connaître la largeur de la seconde dimension.
- `void print(int mat[][3], int lignes, int colonnes)`

# Matrice et vecteurs

- Une matrice est un vecteur de vecteurs.
- Chaque composante d'une matrice (chaque ligne) est un vecteur

```
int M[5][10];
```

```
affiche (M[2], 10);
```



*M[2] est un vecteur de 10 cases*

```
void affiche(int l[], int size)
{
    for(int i=0; i<size; i++)
        cout << l[i] << endl;
    cout << endl
}
```

# Remarques

- On peut créer des tableaux avec **autant de dimensions que l'on veut**.

ex : `century [100] [365] [24] [60] [60]`

# Exercices

80, 83, 81, 82

+ exercices supplémentaires