

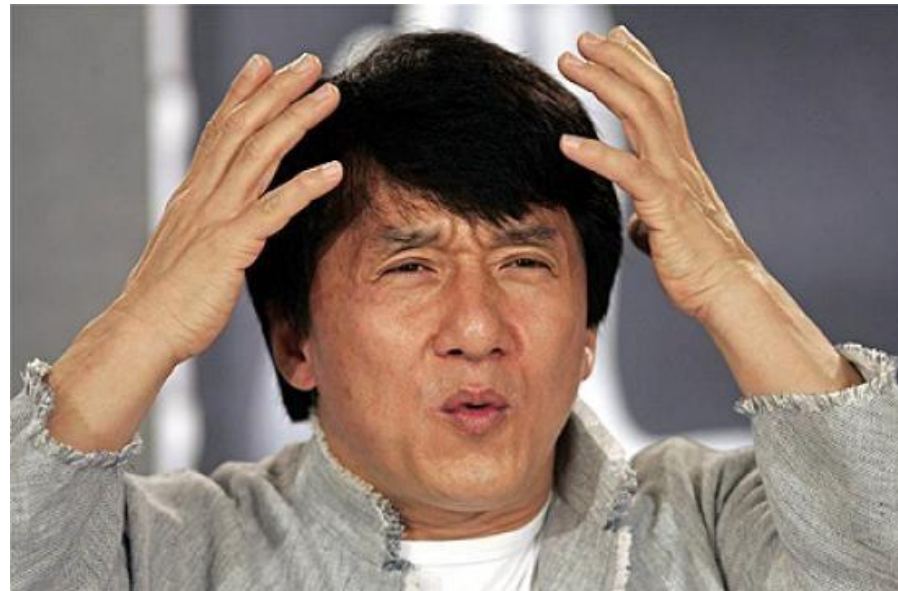
**Tableaux  
multidimensionnels**

# Tableau multidimensionnel

Un tableau à 2 dimensions est un tableau de tableaux.

Un tableau à 3 dimensions est un tableau de tableaux de tableaux.

Un tableau à 4 dimensions est un tableau de tableaux de tableaux de





# Déclaration

```
type nom_tableau[taille1][taille2]...[tailleN];
```

Exemple :

```
int coordinates[3][2] = {{1, 2}, {3, 4}, {5, 6}};
```

est un tableau à 2 dimensions contenant 3 tableaux de 2 éléments.

```
int days[2025][12][31];
```

pourrait être un tableau qui contiendrait une case pour chaque jour depuis JC !





# Accès aux valeurs

```
int days[2023][12][31];
```

// Pour afficher la 7<sup>ème</sup> case du tableau contenu dans  
la 9<sup>ème</sup> case du 1984<sup>ème</sup> tableau :

```
printf("%d", days[1983][8][6]);
```

# Accès aux valeurs

// Initialisation d'un tableau de 2 lignes et 4 colonnes

```
int tableau[2][4];
```

```
for (int i = 0; i < 2; i++) {  
    for (int j = 0; j < 4; j++) {  
        tableau[i][j] = 4 * i + j + 1;  
    }  
}
```

1	2	3	4
5	6	7	8

# Accès aux valeurs

```
int tableau[2][4];
```

1	2	3	4
5	6	7	8

Quel est le type de tableau ?

int \* \*

Quel est le type de tableau[0] ?

int \*

Quel est le type de tableau[0][0] ?

int

Ecrire en notation pointeur tableau[1][3]

\*(\*(tableau + 1) + 3)



**Paramètres  
d'entrée**

**Avez-vous déjà remarqué  
ce genre de paramètres ?**

Sécurité Détails Versions précédentes

meined - Chrome

---

Type de cible : Application

Emplacement : Application

Cible : `ogle\Chrome\Application\chrome.exe --profile-directory="Profile 6"`

---

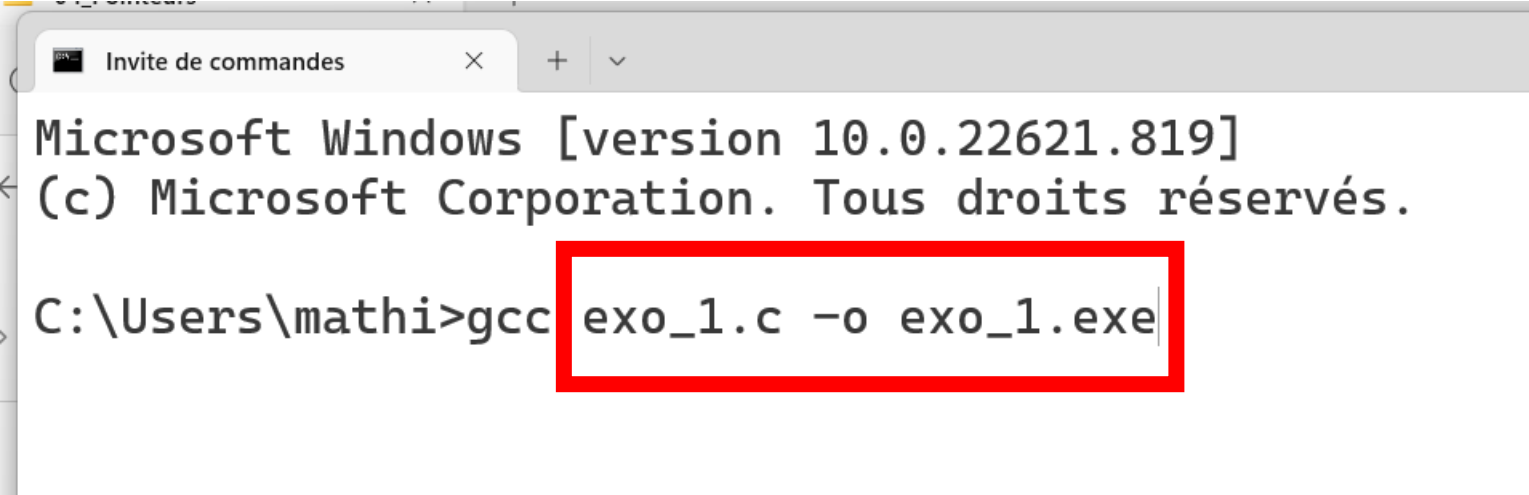
Démarrer dans : `"C:\Program Files (x86)\Google\Chrome\Application"`

Touche de raccourci : Aucun

Exécuter : Fenêtre normale

Commentaire : Accéder à Internet





```
Microsoft Windows [version 10.0.22621.819]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\mathi>gcc exo_1.c -o exo_1.exe
```

## Configuration de Git

Comme vous avez pu l'entrevoir dans [Démarrage rapide](#), vous pouvez s configuration de Git avec la commande `git config`. Une des première été de paramétrer votre nom et votre adresse de courriel :

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

À présent, vous allez apprendre quelques-unes des options similaires le paramétrer votre usage de Git.

# Paramètres du main()

```
int main() {  
    ...  
}  
      ↗  
int main(int argc, char ** argv) {  
    ...  
}
```

<pre>int argc (c = count)</pre>	<pre>char ** argv ou char * argv[] ou char argv[][] (v = values)</pre>
Le nombre de "mots" de la ligne de commande	La liste des chaînes de caractères de la ligne de commande

# Affichage des données fournies

```
#include <stdio.h>
```

```
int main(int argc, char ** argv) {  
    int i;  
    for (i = 0; i < argc; i++) {  
        printf("%d : %s\n", i, argv[i]);  
    }  
    return 0;  
}
```

# Résultat

```
C:\projects\mathieu\bts1_20-21\02_C\04_Pointeurs>a.exe  
0 : a.exe  
1 : hello  
2 : world  
3 : !
```

argv			
a.exe	hello	world	!
0	1	2	3

**argc = 4**

```
C:\projects\mathieu\bts1_20-21\02_C\04_Pointeurs>a.exe  
0 : a.exe  
1 : hello world !
```

argv	
a.exe	"hello world !"
0	1

**argc = 2**

⚠ Les paramètres entre " " sont considérés comme une seule chaîne de caractère !