

A decorative graphic on the left side of the slide, consisting of a network of thin, light-blue lines and small circles, resembling a circuit board or a neural network, extending vertically from the top to the bottom.

# COURS PROGRAMMATION : DES POINTEURS ET DES TABLEAU

CHARLES 'ARYS' YAICHE

# RÉCAP

- La mémoire
- Les tableau statique en C
- Les chaines de caractères

# PROBLÈME

- Supposons que je veuille écrire une fonction à laquelle on envoie un nombre de minutes. Celle-ci renverrait le nombre d'heures et minutes correspondantes :
  - si on envoie 45, la fonction renvoie 0 heure et 45 minutes ;
  - si on envoie 60, la fonction renvoie 1 heure et 0 minutes ;
  - si on envoie 90, la fonction renvoie 1 heure et 30 minutes.

# MÉMOIRE D'UNE VARIABLE



```
1 int age = 10;
```

```
1 printf("La variable age vaut : %d", age);
```

Pour afficher l'adresse d'une variable on doit utiliser le symbole &

Ex : &age -> l'adresse de la variable

Pour afficher une adresse sur printf, il faut utiliser %p

```
1 printf("L'adresse de la variable age est : %p", &age);
```

```
L'adresse de la variable age est : 0023FF74
```

Adresse	Valeur
0	145
1	3.8028322
2	0.827551
3	3901930
...	...
3 448 765 900 126 (et des poussières)	940.5118

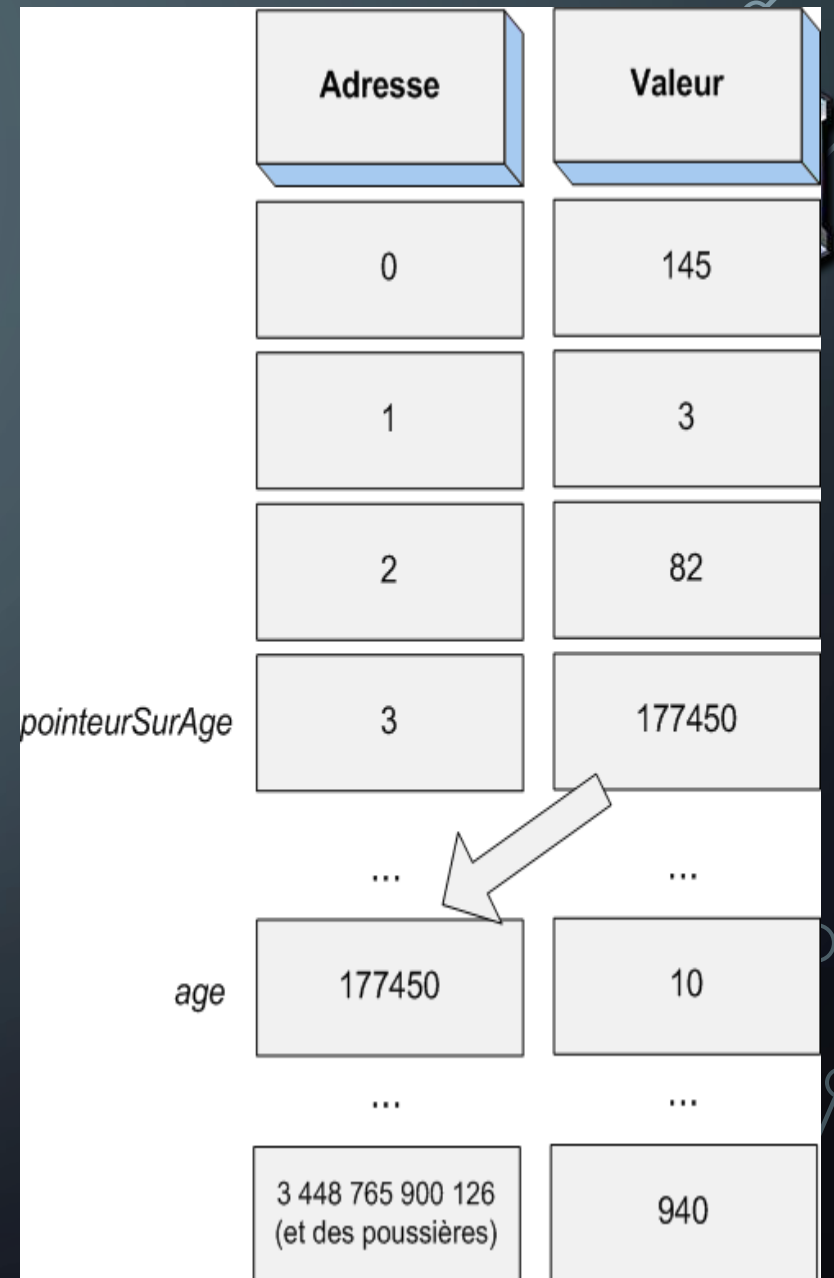
# POINTEUR

- Un pointeur est une variable dont la valeur est une adresse.
- Pour créer un pointeur, il faut utiliser le symbole `*` devant le nom de la variable
- la valeur par défaut n'est pas 0 mais utilise le mot-clé `NULL` (inscrit dans la bibliothèque `stdlib.h`)

```
1 int *monPointeur = NULL;
```

```
1 int age = 10;  
2 int *pointeurSurAge = &age;
```

```
1 int age = 10;  
2 int *pointeurSurAge; // 1) signifie "Je crée un pointeur"  
3 pointeurSurAge = &age; // 2) signifie "pointeurSurAge contient l'adresse de la variable age"
```



# MANIPULER UN POINTEUR

- Pour afficher le contenu de la variable pointé par un pointeur, il faut utiliser le symbole \* devant

```
1 int age = 10;  
2 int *pointeurSurAge = &age;  
3  
4 printf("%d", *pointeurSurAge);
```

```
10
```

# CE QUE VOUS DEVEZ RETENIR

Voici ce qu'il faut avoir compris et ce qu'il faut retenir :

- sur une variable, comme la variable « age »:
  - « age » signifie : « Je veux la valeur de la variable « age » »
  - « &age » signifie : « Je veux l'adresse à laquelle se trouve la variable «age» »
- sur un pointeur, comme « pointeurSurAge »:
  - «pointeurSurAge» signifie : « Je veux la valeur de «pointeurSurAge» » (cette valeur étant une adresse)
  - «\*pointeurSurAge» signifie : « Je veux la valeur de la variable qui se trouve à l'adresse contenue dans «pointeurSurAge» »

Adresse	Valeur
0	145
1	3
2	82
3	177450 <i>pointeurSurAge</i>
...	...
177450 <i>&amp;age</i>	10 <i>age</i> <i>*pointeurSurAge</i>
...	...
3 448 765 900 126 (et des poussières)	940

# POINTEUR EN PARAMÈTRE

```
void triplePointeur(int *pointeurSurNombre);

int main(int argc, char *argv[])
{
    int nombre = 5;

    triplePointeur(&nombre); // On envoie l'adresse de nombre à la fonction
    printf("%d", nombre); // On affiche la variable nombre. La fonction a directement modifié
// la valeur de la variable car elle connaissait son adresse

    return 0;
}

void triplePointeur(int *pointeurSurNombre)
{
    *pointeurSurNombre *= 3; // On multiplie par 3 la valeur de nombre
}
```

```
1 void triplePointeur(int *pointeurSurNombre);
2
3 int main(int argc, char *argv[])
4 {
5     int nombre = 5;
6     int *pointeur = &nombre; // pointeur prend l'adresse de nombre
7
8     triplePointeur(pointeur); // On envoie pointeur (l'adresse de nombre) à la fonction
9     printf("%d", *pointeur); // On affiche la valeur de nombre avec *pointeur
10
11     return 0;
12 }
13
14 void triplePointeur(int *pointeurSurNombre)
15 {
16     *pointeurSurNombre *= 3; // On multiplie par 3 la valeur de nombre
17 }
```



# EXERCICE

- Swap
- Résolution du problème

# LA VÉRITÉ SUR LES TABLEAUX ET ARITHMÉTIQUE DES POINTEURS

- En réalité, les tableau ne sont que des sur le premier élément du tableau.



# LES CHAINES DE CARACTÈRES

- Les **chaîne de caractères** sont en réalité codé comme des tableaux de caractères
  - Les chaînes de caractères sont créées avec " "hello world" est une **chaîne de caractères** (ou String)

Adresse	Valeur	Index
18000	'S'	0
18001	'a'	1
18002	'l'	2
18003	'u'	3
18004	't'	4
18005	'\0'	5

# SOURCES

- <https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/15417-a-lassaut-des-pointeurs>