

# Opérations sur les pointeurs

## Introduction

Salut et bienvenu dans cette vidéo consacrée aux opérations qu'on peut appliquer sur les pointeurs. Plusieurs opérations peuvent être appliquées sur les pointeurs, il s'agit de :

L'addition, la soustraction, l'incrément et la décrémentation. Aussi, plusieurs opérateurs peuvent être utilisés avec les pointeurs : &, \*, +, - et [], ==, =, <,>, >=, <=, !=.

## Exemple 1

Soit l'exemple suivant. Quel est le résultat de l'exécution de chaque instruction :

1. `int A[10];`

2. `int *P;`

3. `P = A+9;`

4. `P = A+10;`

5. `P = A+11;`

6. `P = A-1;`

**Clic** Dans l'instruction 1, nous déclarons un tableau de 10 entiers.

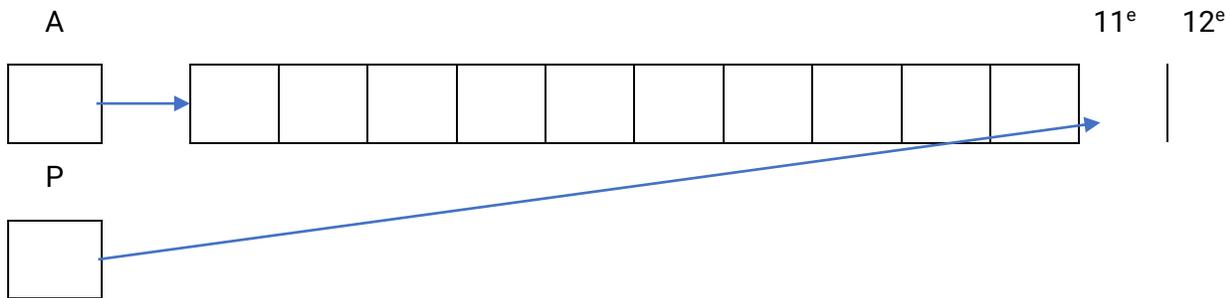
**Clic** Dans l'instruction 2, nous déclarons un pointeur nommé P.

**Clic** Dans l'instruction 3, nous affectons au pointeur P l'adresse du dixième élément du tableau.

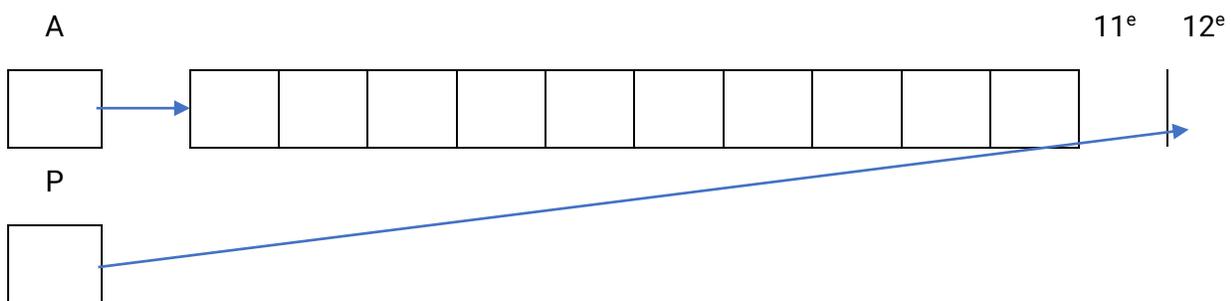
**Clic** L'instruction `A+9` est équivalente à `&A[9]`. Ainsi, l'instruction 3 peut être écrite de deux façons :

1. `P = A + 9`
2. Ou `P = &A[9]`

**Clic** Dans l'instruction 4, nous affectons au pointeur P l'adresse du onzième élément du tableau. Cependant le onzième élément n'existe pas. Il s'agit d'une erreur sémantique, une erreur non logique. Le compilateur ne va générer une erreur lors de la compilation. L'adresse qui sera affectée au pointeur P est l'adresse de l'octet qui vient après le dixième élément du tableau comme indiqué sur le schéma. Ne pas oublier que maintenant le pointeur pointe en dehors du tableau.

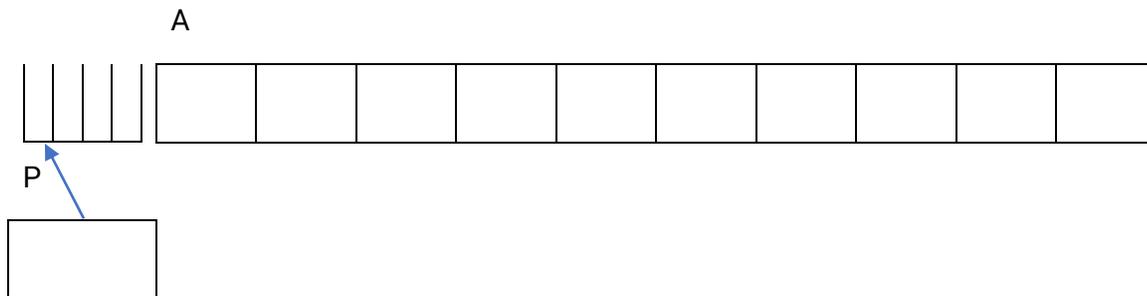


**Clic** Même chose dans l'instruction 5, nous affectons au pointeur l'adresse du douzième élément du tableau. Cependant le douzième élément n'existe pas. Il s'agit d'une erreur sémantique, une erreur non logique. Le compilateur ne va générer une erreur lors de la compilation. Ainsi, l'adresse physique qui sera affectée est l'adresse de l'octet 5 qui vient après le dernier élément du tableau. Les 4 octets qui viennent après le dixième élément du tableau occupent 4 octets dans la RAM et l'octet suivant désigne A + 11 comme indiqué sur le schéma. Ne pas oublier que maintenant le pointeur pointe en dehors du tableau.



Dans l'instruction 6, à partir de l'adresse du premier élément du tableau A, nous revenons en arrière de 4 octets et nous affectons l'adresse de premier octet au pointeur P,

Clic comme indiqué sur le schéma. Ne pas oublier que maintenant le pointeur pointe en dehors du tableau.



## Exemple 2

Soit l'exemple suivant,

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};
```

```
int *P;
```

```
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions :

1. `*P+2`
2. `*(P+2)`
3. `&P+1`
4. `&A[4]-3`
5. `A+3`
6. `&A[7]-P`
7. `P+(*P-10)`
8. `*(P+*(P+8)-A[7])`

Dans expression 1, le résultat est une valeur et non une adresse.

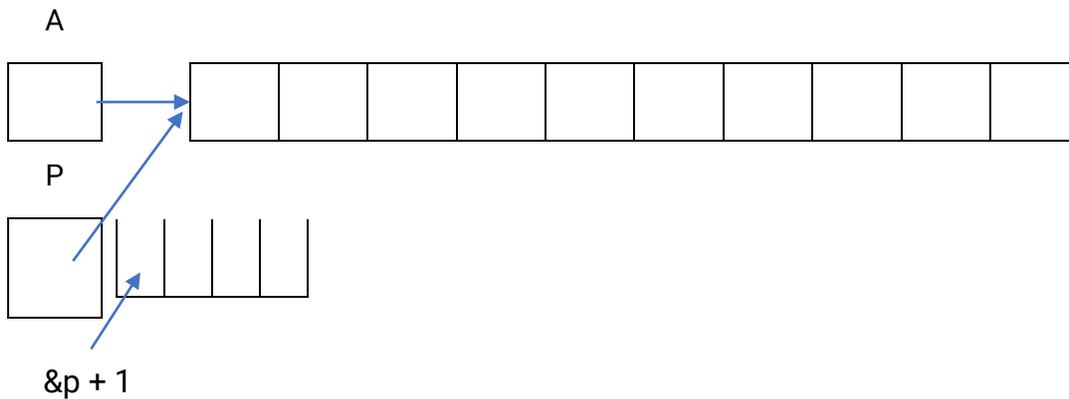
Clic Nous aurons la valeur 14. Pourquoi ? \*p indique la valeur du 1<sup>er</sup> élément du tableau A, il s'agit de 12. Cette dernière sera ajoutée à 2, ce qui nous donne 14.

Dans l'expression 2, ici aussi, nous aurons une valeur et non une adresse.

**Clic** Il s'agit du résultat 34. Pourquoi ?  $*(P+2)$  correspond à  $A[2]$ , c'est à dire à la valeur 34

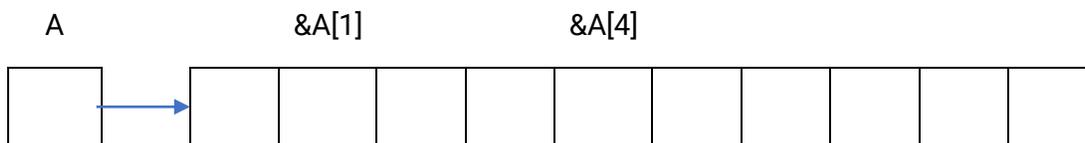
Dans l'expression 3, il s'agit d'un résultat adresse et non d'une valeur entière.  $\&P$  correspond à l'adresse du pointeur P dans la RAM.  $\&P + 1$  correspond à l'adresse du premier octet qui vient après le pointeur P. le Pointeur occupe 4 octets. Et  $\&P$  correspond à l'adresse de son premier octet.

**Clic**  $\&P + 1$  correspond à l'adresse de l'octet 5 qui vient après le pointeur P comme indiqué sur le schéma suivant.



Dans l'instruction 4, le résultat est une adresse.  $\&A[4]$  correspond à l'adresse du 5<sup>ème</sup> élément du tableau. Ainsi  $\&A[4] - 3$  indique qu'il faut revenir en arrière de 3 éléments à partir du 5<sup>ème</sup> élément. Ce qui nous donne l'adresse du 2<sup>ème</sup> élément du tableau c'est à dire  $\&A[1]$ . Pour simplifier l'opération arithmétique de  $\&A[4] - 3$ , il suffit d'écrire  $\&A[4 - 3]$ .

**Clic** Vous remarquez que nous avons exprimé la soustraction entre crochet :  $\&A[4-3] \rightarrow \&A[1]$ .



L'expression 5 fournit un résultat adresse.  $A$  correspond à l'adresse du 1<sup>er</sup> élément du tableau A, c'est-à-dire  $\&A[0]$ .

**Clic**  $A+3$  est équivalent à  $\&A[0] + 3$ , il suffit maintenant d'exprimer l'opération arithmétique l'addition entre crochet :  $\&A[0 + 3]$  ce qui nous donne  $\&A[3]$  l'adresse du 4<sup>ème</sup> élément du tableau.

Dans l'expression 6, nous avons  $&A[7]-P$ . Nous pouvons écrire cette expression de la façon suivante : **Clic**

$&A[7]-&A[0]$ , nous avons remplacé  $P$  par l'adresse du premier élément du tableau  $&A[0]$ . Cette expression donne le résultat comme suit :  $7 - 0$  qui est égale à 7.

Dans l'expression 7 nous avons  $P+(*P-10)$ . Nous allons évaluer cette expression Evaluons d'abord  $(*P-10)$ ,  $*P$  correspond à  $A[0]$  ainsi  $A[0] - 10 = 12 - 10$  ce qui nous donne 2.

**Clic**  $P + 2$  correspond à  $&A[2]$  c'est-à-dire l'adresse du 3<sup>ème</sup> élément du tableau A.

Dans l'expression 8, nous avons l'expression suivante :  $*(P+*(P+8)-A[7])$ .

**Clic** Pour l'évaluer, nous allons d'abord évaluer  $*(P+8)-A[7]$ .  $*(P+8)$  correspond à  $A[8]$ . Ainsi l'expression  $*(P+8)-A[7]$  peut être écrite comme suit :  $A[8]-A[7] = 90 - 89 = 1$ .

**Clic** L'expression  $*(P+*(P+8)-A[7])$ , peut être écrite comme suit :  $*(P + 1)$  et cette dernière correspond au 2<sup>ème</sup> élément du tableau A[1] c'est-à-dire la valeur 23.