

## Série 10: pointeur (solution)

Lien avec le [MOOC Initiation à la Programmation \(en C++\)](#)

### Exercices semaine7 du MOOC : pointeur

#### Exercice complémentaire (ExC)

##### ExC 6: évaluation de code sans le compiler (niveau 1)

**p1** pointe sur **a**, **p2** et **p3** pointent sur **b**. Les affectations avec **indirection** sur **p2** ou sur **p3** à gauche de l'opérateur '=' modifient **b**.

De même l'affectation avec une **indirection** sur **p1** à gauche de l'opérateur '=' modifie **a**.

Les valeurs finales sont:

**a: 27**

**b: 20**

##### ExC 7 : Type et valeur d'expressions avec des pointeurs et un tableau à-la-C (niveau 1)

a) La valeur de l'expression `*ptr_1 + *ptr_2` est 12.

La valeur de l'expression `c [1] == *ptr_2` est 1.

La valeur de l'expression `*c + b` est 3.

La valeur de l'expression `*(c+b)` est 7.

Celle de l'expression `ptr_1 == ptr_2 ? 2 * *ptr_1 : 3 * *ptr_2;` est 6.

b) après exécution de l'instruction `*ptr_1 = *ptr_2 + a`

la valeur de la variable **a** est 12 et la valeur de **b** ne change pas.

c) Le type de l'expression `&ptr_1` est `int **`.

Le type de l'expression `*ptr_1` est `int`.

d) L'instruction `c++`; n'est pas valide, car **c** est un pointeur constant ; il ne peut pas être à gauche d'un symbole d'affectation. On dit encore que **c** n'est pas une *lvalue* (comme *left-value* parce qu'elle est à gauche de l'opérateur d'affectation). Le compilateur produira un message d'erreur.

Par contre `ptr_3++`; est valide.