# Série 10: pointeur

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

# Exercices semaine7 du MOOC : pointeur

- **Document Exercices semaine 7 du MOOC**

  - **Exercice 21 : généricité (niveau 1)**
    - Premier exemple d'application de la notion de pointeur

  - **Exercice 22 : référence (structures, références, niveau 1)**
    - Intérêt et limitation des références

# Exercice complémentaire (ExC)

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

```
#include <iostream>
using namespace std;

int main()
{
    int a(12);
    int b(34);

    int *p1(nullptr);
    int *p2(nullptr);
    int *p3(nullptr);

    p1 = &a;
    p2 = &b;

    p3 = p2;

    *p2 = 15;
    *p1 = (*p1) +  (*p3);
    *p3 = (*p2) + 5;

    cout << a  << " " << b << endl;

    return 0;
}
```

Soit le morceau de programme ci-contre. Les valeurs de **a** et **b** sont elles modifiées pendant l'exécution de ce programme?

si oui quelles sont leurs valeurs finales.

Pour appuyer votre réponse, dessiner les liens entres les variables avec la représentation par boite (pour les variables) et flèche (lorsqu'un pointeur pointe sur une variable).

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

Soient les déclarations suivantes:

```
int a(10), b(2);
int c[3] = { 1, 2, 7 };

int *ptr_1(&a);
int *ptr_2(&b);
int *ptr_3(c); // le nom du tableau c (tableau à-la-C)
               // est un pointeur constant sur son premier élément
```

a) Quelle est la valeur des expressions suivantes (faire un dessin boite/flèche pour illustrer l'état des variables):

- **`*ptr_1 + *ptr_2`**
- **`c [1] == *ptr_2`**
- **`*c + b`**
- **`*(c+b)`**
- **`ptr_1 == ptr_2 ? 2 * *ptr_1 : 3 * *ptr_2;`**

b) Quelle est la valeur des variables a et b après exécution de l'instruction ***ptr_1=*ptr_2 + a;**

c) Quel est le type de l'expression &ptr_1 ?

Même question pour l'expression *ptr_1

d) L'instruction c++; est-elle valide ?

Même question pour l'instruction ptr_3++;