

Méthodologie de conception et de programmation_S2_P

[Mon espace](#) / [Cours](#) / [Sciences et Technologies](#) / [Faculté des Sciences et Technologies \(FST\)](#)
/ [MIAE](#) / [Licence Informatique](#) / [L1 Semestre S2](#)
/ [Méthodologie de conception et de programmation_S2_P](#)
/ [Pointeurs --- Mémoire d'un processus --- Tableaux \(chapitre 3\)](#) / [Exercices sur les pointeurs](#)

Exercices sur les pointeurs

Exercice 1 :

Dans le `main` : déclarer deux entiers `i` et `j`.

Affecter à `i` la valeur 3.

Affecter à `j` la valeur de `i`.

Afficher les valeurs de `i` et `j`.

Afficher la taille en octet de `i` et `j` (fonction `sizeof` qui retourne un `size_t` et spécification de format pour un `size_t` à afficher en entier décimal : `%zd`).

Afficher les adresses de `i` et `j` (spécification de format pour une adresse : `%p`).

Déclarer `p` un pointeur sur un entier. Le faire pointer vers l'entier `i`.

Afficher la valeur de `p`.

Afficher son adresse.

Afficher la valeur de l'entier vers lequel il pointe.

Afficher la valeur de `i`.

Modifier `i` : lui affecter la valeur 42 par exemple.

Reprendre les affichages ci-dessus. Que constatez-vous ? Est-ce surprenant ?

Modifier `*p` (la valeur pointée par `p`) : lui affecter la valeur 154 par exemple.

Reprendre les affichages ci-dessus. Que constatez-vous ? Est-ce surprenant ?

Exercice 2 :

Déclarer deux entiers `i` et `j` qui valent respectivement 7 et 8.

Déclarer deux pointeurs vers des entiers `p1` et `p2`.

Affecter à `p1` l'adresse de `i` et affecter à `p2` l'adresse de `j`.

Afficher les valeurs de `i`, `j`, `p1`, `p2`, `*p1` et `*p2`.

Affecter à `*p1` la valeur de `*p2`.

Afficher les valeurs de `i`, `j`, `p1`, `p2`, `*p1` et `*p2`. Expliquer ce qui s'est passé.

Affecter à `*p1` la valeur 10.

Affecter à `p1` la valeur de `p2`.

Afficher les valeurs de `i`, `j`, `p1`, `p2`, `*p1` et `*p2`. Expliquer ce qui s'est passé.

Exercice 3 : Arithmétique des pointeurs

Déclarer un tableau d'entiers à deux cases `exo3` qui contient `{3, -3}`.

Déclarer deux pointeurs vers des entiers `p1` et `p2`.

Affecter à `p1` l'adresse de `exo3[0]` et à `p2` la valeur de `p1 + 1`.

Afficher les valeurs de `p1` et `p2`.

Afficher la valeur de `p2 - p1` et afficher la taille en octet d'un `int`.

Même chose avec des `double`.

Exercice 4 : Allocation dynamique

Déclarer un pointeur `p` vers un entier.

Afficher la valeur de `p`.

Affecter à `*p` la valeur 3.

Afficher la valeur de `*p`.

Compiler et exécuter : que se produit-il ? Indice : « ça ne marche pas ! »

Vous avez oublié d'allouer de la mémoire pour stocker la valeur 3.

```
p = (int*)malloc(sizeof(int));
```

Corriger votre programme.

N'oublier pas d'utiliser l'instruction `free` à la fin du programme : `free(p)` ;

Compiler et exécuter : que se produit-il ?

Déclarer un entier `i` initialisé à 5.

Affecter à $*p$ la valeur de i .

Afficher les adresses et les valeurs de i , p et $*p$. Expliquez ce qui s'est passé.

Lors de l'affectation de la valeur de p , allouer maintenant l'espace pour 2 entiers.

Affecter à $*(p+1)$ la valeur 42.

Afficher les adresses et les valeurs de i , p et $*p$, $(p+1)$ et $*(p+1)$. Expliquez ce qui s'est passé.

Modifié le: vendredi 30 janvier 2015, 17:43

[Méthodo_L1_info_S2_P](#) |

Outils enseignant

[Gestion des cours](#)

[Calendrier des formations](#)

Français (fr)

[Deutsch \(de\)](#)

[English \(en\)](#)

[Français \(fr\)](#)

[Mentions légales](#)