

Exercice 307

Enoncé

Soit les instructions suivantes :

```
// 0. Déclaration des variables
int compteur, indice;
// 1. Début du code
compteur = 10;
indice = compteur;
++compteur;
compteur++;
indice*=2;
compteur = ++compteur *4;
indice = indice*7+45/5;
compteur = compteur*(3-54/6);
```

- Quelle différence entre les instructions `compteur++` et `++compteur` ?
- Réécrire l'instruction : `indice*=2` ;
- Faire un schéma mémoire de ces instructions et donner la valeur de ces variables

Corrigé

Le code se trouve dans le fichier `main.c`.

Différence entre `compteur++` et `++compteur`

Les deux utilisations du `++` incrémentent de 1 la variable. La différence se situe dans la valeur que renvoie l'opération. Prenons ce code:

```
int i = 0;
int variable;
variable = ++i; // Première affectation
i = 0;
variable = i++; // Seconde affectation
variable = i; // Troisième affectation
```

Lors de la première affectation, `variable` vaut 1. Lors de la seconde, `variable` vaut 0. Et lors de la troisième, `variable` vaut 1.

`++i` est un pré-incrément. La variable est d'abord incrémentée, puis rendue. (D'où la valeur 1)

`i++` est un post-incrément. La variable est d'abord rendue, puis incrémentée. (D'où la valeur 0). La variable est bien incrémentée, puisque lors de la troisième affectation, elle vaut 1.

Réécrire l'instruction : `indice*=2 ;`

L'opérateur `*=` est un opérateur d'affectation. Il est équivalent à :

```
indice = indice * 2;
```

On peut aussi le faire avec d'autres opérateurs :

Opérateur	Équivalent
<code>+=</code>	<code>var = var + 2;</code>
<code>-=</code>	<code>var = var - 2;</code>
<code>/=</code>	<code>var = var / 2;</code>
<code>%=</code>	<code>var = var % 2;</code>

Schéma Mémoire

Pas de schéma mémoire pour l'instant.

```
// 0. Déclaration des variables
int compteur, indice;
// 1. Début du code
compteur = 10;           // compteur = 10
indice = compteur;      // indice = 10
++compteur;             // compteur = 11
compteur++;             // compteur = 12
indice*=2;              // indice = 20
compteur = ++compteur *4; // compteur = 52
indice = indice*7+45/5;  // indice = 149
compteur = compteur*(3-54/6); // compteur = -312
```

Commentaires d'exercice

L'opération `compteur = ++compteur *4;` n'est pas fausse, mais peut amener à une mauvaise compréhension de l'opérateur `++`. `++compteur` modifie directement la valeur de `compteur`. L'assigner à `compteur` ne sert à rien. De plus une variante simple de cela : `compteur = compteur++ * 4` a un comportement indéfini. C'est à dire que selon les machines et les compilateurs, le résultat peut être différent. Il est donc préférable de ne pas utiliser ce genre de code.