Exercice 301

Enoncé

- Combien y a-t-il de types scalaires en C ? Nommez-les et donner leur plage de valeurs ?
- Par défaut, sont-ils signés ou non signés ?
- Quelle est la place de la déclaration des variables locales dans un code bien écrit ?
- Comment déclarer une variable ?
- A quoi sert l'indentation d'un programme ?

Corrigé

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

Types scalaires en C

Il existe une multitude de types scalaires. On a par exemple les chars et les ints :

Type	Plage de valeurs
char	-128 à 127
int	-2^31 à 2^31 - 1

Ce ne sont cependant qu'une petite partie des types qui existent. Il existe par exemple des types de variables que l'on qualifie de "non signé". Leur avantage est d'effectivement doubler la valeur maximale que peut stocker un type mais en lui retirant la capacité de stocker des valeurs négatives.

Explication technique d'une variable non-signée

Un int classique, par exemple, 346 ressemble à ceci en binaire:

```
00000000 00000000 00000001 01011010
```

Le même nombre, mais négatif cette fois ci ressemble à ceci:

```
10000000 00000000 00000001 01011010
```

Remarquons que la seule différence est dans le bit de poids fort, qui permet de désigner le signe. On obtiens donc la capacité à différencier les nombres positifs des nombres négatifs. Cependant, pour un int par exemple, nous avons maintenant 2^31 valeurs de chaque côté de 0. (En réalité pour pouvoir aussi stocker '0', on ne peut stocker que 2^31 - 1 valeurs positives et 2^31 valeurs négatives).

Voici d'autres types et leurs plages de valeurs :

Type	Plage de valeurs
char	-128 à 127
char non signé	0 à 255
short	-32768 à 32767
short non signé	0 à 65535
int	-2^31 à 2^31 - 1
int non signé	0 à 2^34
long int	2^63 à 2^63 - 1
long int non signé	0 à 2^64

Par défaut, une déclaration de variable est signée.

Déclaration d'une Variable et Variables Globales

Un code bien écrit ne contient, de facto, pas de variables globales. Des variables globales sont des variables qui ne se trouvent pas à l'intérieur du `main()` ou d'un autre sous-programme. Exemple:

```
#include <stdio.h>

int maVariable; // Cette variable est une variable globale

int main() {

    int maVariable2; // Cette variable est une variable locale au main()

    printf("Hello World");

    return 0;
}
```

Elles sont dites 'globales' car elles sont accessible par tout le code du fichier. Elles sont reconnues par la globalité du code. En revanche une variable dans le `main()` est 'locale', car elle ne peut être accédée que par ce dernier.

Variables Globales, Why Not ?

Les variables globales sont à éviter à tout prix pour plusieurs raisons. Pour des petits exercices qui peuvent être terminés en un fichier unique et plutôt court, elles ne sont peut-être pas immédiatement évidentes. Il faut donc bien faire attention à ne pas prendre de mauvaises habitudes.

D'une part, une variable globale est comparable à un joker. Très puissante et potentiellement utile, mais imprévisible. Tout grand projet de nos jours met en place le principe de 'unit testing', c'est à dire que pour chaque petite fonctionnalité, on implémente un test qui va tester tous les cas de figures et vérifier que la dite petite fonctionnalité ne s'effondre pas devant des entrées inattendues (par exemple, si le programme

s'attend à un entier, et qu'on lui fournit un texte, il faut qu'il puisse réagir de manière correcte, sans forcément aboutir à un crash - en redemandant une entrée par exemple). Si la fonctionnalité à tester utilise une variable globale, le test ne pourra pas modifier cette variable à son gré (car elle est potentiellement utilisée quelque part d'autre) et ne pourra pas conclure sur la fiabilité de la fonctionnalité.

D'autre part, une variable globale vaut -2 points sur un rendu selon le professeur. Elles sont donc à éviter.

Elles sont aussi très difficiles à débuguer car, étant modifiable par tout le code, il est plutôt difficile de remonter à la source du problème.

Comment Déclarer une Variable

Déclarer une variable est la première étape pour pouvoir stocker une valeur. (A noter que tout se stockera en tant que nombre. Même les caractères. Explications dans le chapitre suivant)

Pour déclarer une variable en C, on indique d'abord son type, puis son nom :

```
int maVariable;
```

Cette variable va donc servir à stocker un entier (`int`) et se nomme `maVariable`

Indentation

En C, le formatage est facultatif. On peut parfaitement écrire tout son code sur une seule ligne. Cela veut dire que l'indentation et le formatage sont simplement une question visuelle. Pour l'indentation, elle permet de facilement savoir quels éléments sont locaux à quels autres éléments.

Par exemple :

```
int main(){

    int i = 5;
    if(i = 5)
    {
        int j = 8;
    }
    else
    {
        int k = 54;
    }

    return 0;
}
```

Ici, dans ce code sans utilité, on voit rapidement que i est local à la fonction `main()`, j à `if()` et k à `else`.

Remarque : on ne peut pas accéder à j en dehors des accolades (`{}`) du `if()`, ni à k en dehors du `else`.