

### 2.1.6 Les variables globales

Imaginons que nous voulions isoler l'instruction `x = 0 ;` dans le programme

```
int x;  
x = 3;  
x = 0;
```

Cela nous amènerait à écrire la fonction

```
static void reset () {x = 0;}
```

puis le programme principal

```
int x;  
x = 3;  
reset();
```

Mais ce programme est incorrect, car l'instruction `x = 0 ;` n'est plus dans la portée de la variable `x`. Pour que la fonction `reset` puisse avoir accès à la variable `x`, il faut déclarer la variable `x` comme une variable *globale*, c'est-à-dire commune à toutes les fonctions du programme et au programme principal

```
static int x;
```

```
static void reset () {x = 0;}
```

puis le programme principal

```
x = 3;  
reset();
```

Toutes les fonctions peuvent utiliser toutes les variables globales, que celles-ci soient déclarées avant ou après la fonction.

### 2.1.7 Le programme principal

Un *programme* est donc constitué de la déclaration de variables globales  $x_1, \dots, x_n$ , puis de la définition de fonctions  $f_1, \dots, f_{n'}$ , puis du *programme principal* `p` qui est une instruction.

Un programme pourrait donc s'écrire

```
static T1 x1 = t1 ;  
...  
static Tn xn = tn ;
```