

Comme en mathématiques, si on veut diviser deux nombres, on les sépare par le symbole `/`

Tout se passe bien si on divise deux nombres à virgules

```
cout << (42.0 / 4.2) << endl;
```

↳ 10.0

ou bien un nombre à virgule et un entier

```
cout << (42.0 / 2) << endl;  
cout << (42 / 2.0) << endl;
```

↳ 21.0
21.0

Par contre, si on cherche à diviser deux entiers, on obtient ceci

```
cout << (42 / 10) << endl;
```

↳ 4

ce qui est plutôt surprenant !

Explications

A partir du moment où un nombre décimal est impliqué dans la division, le résultat sera toujours un nombre décimal. Par contre, **si les deux nombres sont des entiers le résultat sera un entier**.

Ainsi `42 / 10` vaut, mathématiquement parlant, `4.2` mais comme le résultat de la division d'un entier par un entier est un entier alors ce `4.2` est transformé en `4`, en oubliant tout ce qu'il y a après la virgule.

Il peut arriver que les deux nombres soient des entiers mais que l'on désire que le résultat soit un nombre décimal. Il faut alors convertir un ou les deux nombres en décimaux avant de faire la division. Par exemple, pour convertir le premier nombre en décimal on écrit :

```
cout << ( (double) (42) ) / 10 << endl;
```

↳ 4.2

La syntaxe `((double) (quelqueChose))` permet donc de transformer le `quelqueChose` en un `double`. La division fait alors intervenir un `double` et un `int`, ce qui donnera comme résultat un `double`.