

Quand on nous donne une chaîne de caractères on a parfois besoin de connaître sa longueur, c'est-à-dire le nombre de caractères qu'elle contient. En C++, il existe une fonction pour cela, comme on peut le voir sur le code suivant :

```
string vers = "Que vous etes joli! Que vous me semblez beau!";  
int longueur = vers.length();  
cout << longueur << endl;
```

↳ 45

## Comparer une longueur à un entier

Si on écrit un code code celui-ci

```
string vers = "Que vous etes joli! Que vous me semblez beau!";  
if (vers.length() > 50)  
{  
    cout << vers << endl;  
}
```

on va avoir un warning à la compilation :

↳ *warning: comparison between signed and unsigned integer expressions*

Il se trouve qu'il existe plusieurs types d'entiers en C++, les `int` qui peuvent être négatifs et les `unsigned int` qui ne peuvent être que positifs. Comme une longueur est toujours positive, la fonction `length()` renvoie un `unsigned int`. Le problème est que comparer des `int` et des `unsigned int` peut donner des choses très inattendues, d'où le warning. La raison exacte vous sera expliquée plus tard, quand nous verrons les différents types de données en C++.

Pour ne plus avoir ce warning, il suffit de dire explicitement dans le programme, qu'on souhaite considérer cette longueur comme un simple `int` :

```
string vers = "Que vous etes joli! Que vous me semblez beau!";  
if ((int)(vers.length()) > 50)  
{  
    cout << vers << endl;  
}
```

## Rapidité du calcul d'une longueur

Calculer la longueur d'une chaîne de caractère est une opération **rapide** en C++, qui se fait en temps constant : quelle que soit la longueur de la chaîne, cela prend toujours le même temps.