

Recherche dans les chaînes

Rechercher un caractère

La recherche d'un caractère dans une chaîne peut également être réalisée en utilisant des fonctions membres et des fonctions libres (comme pour les fonctions `begin` et `end` dans le chapitre précédent). C'est le cas de la fonction de base pour faire une recherche : `find` (*trouver* en anglais). D'autres fonctions de recherche ne sont utilisable qu'en fonction membre :

- `rfind` recherche à partir de la fin (*reverse find*) ;
- `find_first_of` recherche la première occurrence d'un caractère ;
- `find_first_not_of` recherche la première absence de caractère ??? ;
- `find_last_of` recherche la dernière occurrence d'un caractère ??? ;
- `find_last_not_of` recherche la dernière absence d'un caractère ???.

D'autres fonctions ne sont utilisables que sous forme de fonction libre (ce sont les algorithmes de la bibliothèque standard) :

- `count` pour compter le nombre d'occurrence d'un caractère ;

Comme pour les algorithmes de la bibliothèque standard vu dans le chapitre précédent (`std::equal` et `std::lexicographical_compare`), il existe également une version de ces algorithmes précédents utilisant un prédicat personnalisé. Cependant, ici, les algorithmes change de nom, mais dans tous les cas, le prédicat est le dernier argument donné.

- `find_if`
- `find_if_not`

- `count_if`

Vérifier qu'une chaîne contient une sous-chaîne

```
string const s { "une petite phrase" };
cout << s.find("petite") << endl;
cout << s.find("totot") << endl;
cout << s.find('w') << endl;

cout << s.rfind("petite") << endl;

// réutilisation de pos
auto pos { s.find('e') }; // trouve le premier e
cout << pos << endl;
cout << s.find('e', pos) << endl;
```

`find_first_of` : 1 caractère dans une liste de caractères

`find_first_not_of` : l'inverse

`find_last_of`

`find_last_not_of`

`parsing` : extraire une information

`parcourir` toutes les correspondances

`token` : split une chaîne en sous chaînes

Chapitre précédent	Sommaire principal	Chapitre suivant
Cours, C++		