

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Exemples simples (vector) (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

Éléments du tableau. Boucle c. Nom du tableau. Tableau dynamique. Nom de variable. Position i. Indices des éléments du tableau. Ancienne valeur du tableau. Nouvelle valeur d'un tableau. Certainne taille. Cadre d'illustrations des itérations. Affectation globale de tableaux. Telle expression. Copie de tab2. Corps de la boucle.



[vers la recherche de séquences vidéo](#)
(dans Initiation à la programmation (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Tableaux : exemples simples (vector)

(Partie 2)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Dans ces deux exemples où l'on affiche les éléments du tableau,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



.....

.....

.....

.....

.....

(ré)Affectation de tous les éléments à la valeur 1.2 :

```
→ for(auto& el : tab) {  
    el = 1.2;  
}
```

ou alors

```
tab = vector<double>(tab.size(), 1.2);
```

on n'a donc pas besoin de modifier les éléments du tableau. Voyons, maintenant, un exemple où l'on souhaite, par exemple, affecter la valeur, à tous les éléments du tableau, la valeur 1.2 Si l'on veut utiliser une boucle C++ 2011, alors on déclarera ici « for(auto » avec le signe pour indiquer que l'on va modifier ('&'), un nom de variable -- j'avais choisi « élément » au préalable, ici, je peux choisir « el » -- le ':' pour séparer les deux parties et donc le nom du tableau. Dans cette boucle-là, on va donc pouvoir utiliser el comme un nom de variable qui va parcourir tour à tour les différents éléments du tableau tab et donc, ici, pour chacun de ces éléments, on va mettre la valeur 1.2 On pourrait aussi dans un tableau dynamique si on voulait affecter directement tout le tableau on peut aussi utiliser, mais ça sort du cadre d'illustrations des itérations, utiliser une affectation globale de tableaux. Ici, on va donc écraser le tableau, l'ancienne valeur du tableau, par la nouvelle valeur d'un tableau qui est, alors, on regarde ici, élément par élément : un tableau dynamique qui contient des double que l'on initialise, au départ,

notes

résumé

0m 5s



(ré)Affectation de tous les éléments à la valeur 1.2 :

```
→ for(auto& el : tab) {  
    el = 1.2;  
}
```

ou alors

```
tab = vector<double>(tab.size(), 1.2);
```

à une certaine taille qui est l'ancienne taille du tableau tab donc on donne tab.size() et dont tous les éléments prennent la valeur 1.2

notes

résumé

1m 13s



(ré)Affectation de tous les éléments à la valeur 1.2 :

```
for(auto& el : tab) {  
    el = 1.2;  
}
```

ou alors

```
tab = vector<double>(tab.size(), 1.2);
```

```
vector<double> tab2(tab.size(), 1.2);  
tab = tab2;
```

Ça crée bien un nouveau tableau dynamique qui a exactement `tab.size()` éléments et dont chacun des éléments a la valeur 1.2. C'est comme si vous aviez déclaré en deux lignes, tout d'abord un tableau `tab2` qui est un tableau dynamique de « double », initialisé à la taille de `tab` donc `tab.size()` et tous à la valeur 1.2 puis qu'ensuite vous ayez fait la copie de `tab2` dans `tab`. On peut l'écrire comme ça, en une seule ligne, de façon compacte, en déclarant ici ce qu'on appelle un tableau anonyme. On n'a pas mis le nom ici où on aurait mis normalement le nom `tab2` si on l'avait déclaré en deux lignes. c'est un tableau sans nom, un tableau anonyme. Une telle expression comme ceci, on peut tout à fait l'écrire c'est un tableau qui existe qui est créé ponctuellement avant d'être recopié dans `tab`.

notes

résumé

1m 25s





Autre exemple de modification typique

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

2m 25s



.....

.....

.....

.....

.....

Saisie au clavier des éléments du tableau :

- ▶ si l'on n'a pas besoin d'explicitement les indices :

```
for(auto& element : tab) {
    cout << "Entrez l'élément suivant :" << endl;
    cin >> element;
}
```



- ▶ si l'on veut expliciter les indices :

```
for(size_t i(0); i < tab.size(); ++i) {
    cout << "Entrez l'élément " << i << " : " << endl;
    cin >> tab[i];
}
```

des éléments d'un tableau, c'est lorsqu'on les demande à l'utilisateur. Là aussi, on a deux cas. Soit on veut afficher explicitement les positions, les indices des éléments du tableau. Soit on souhaite simplement juste les demander, les éléments du tableau les uns après les autres sans expliciter les indices. Regardons ces deux cas. Si on n'a pas besoin d'explicitement les indices alors on peut utiliser une boucle for de type C++ 2011. Mais comme on veut modifier le tableau, puisqu'on va saisir les valeurs, on va utiliser ici une boucle par référence. Donc on retrouve le ':' des boucles C++ 2011 et toujours les deux parties, le tableau que l'on veut modifier tab et puis, ici, la partie déclarative avec le nom qu'on a choisi pour parcourir les éléments du tableau, par exemple, élément. Puis, dans cette boucle, donc le corps de la boucle toujours contrôlé par le bloc ici dans cette boucle, on va afficher « Entrez l'élément suivant », le message par exemple, et puis saisir dans la variable élément. La variable élément va parcourir, encore une fois, tour à tour, les différents éléments du tableau en commençant par le premier puis quand on va faire la 2e itération, le deuxième élément du tableau puis lorsqu'on va reboucler dans la troisième itération on sera sur le troisième élément du tableau, etc, etc jusqu'à arriver au dernier élément du tableau. Si vous voulez expliciter les indices, alors on utilisera une boucle for plus classique séparée en trois parties avec le ; on déclare comme d'habitude ici un entier i de type size_t à la position du premier élément, c'est-à-dire la position 0. Tant que i est strictement inférieur à tab.size()

notes

résumé

2m 28s



on incrémente ici de 1 en 1 et on va donc afficher ici l'indice, c'est ce qu'on voulait, c'est pour ça qu'on a introduit i donc on dit : « Entrez l'élément 0 », « l'élément 1 », etc. Puis, enfin, on va saisir au clavier l'élément à la position i dans le tableau `tab` donc « `tab[i]` » qui désigne cet élément. donc « `tab[i]` » qui désigne cet élément.

4m 13s

