

Support de cours

Cours:

## Initiation à la programmation (en C++)

Vidéo:

### Itérations - quiz (partie 3)

Concepts (extraits des sous-titres générés automatiquement) :

**Première boucle for. Instruction d'incrémentation. Sortir de la boucle. Boucle for. Variable j. Fin de la boucle for. Bonne réponse. Deuxième boucle for. Suite de cette boucle for. Variable i. Fin de la boucle. Instruction for. Fin du code. Instruction cout. Valeur.**



[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/>

# Itérations : quizz

(Partie 3)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Que s'affiche-t-il quand on exécute le code :

```
for(int i(0); i < 3; ++i) {  
    for(int j(0); j < i; ++j) {  
        cout << j;  
    }  
    cout << endl;  
}
```

A:

0  
01

B:

0  
01  
012

C:

rien

D:

0123  
0123  
0123



La bonne réponse, est la réponse A, et nous

notes

---

---

---

---

---

---

---

---

---

---

résumé

0m 1s



---

---

---

---

---

---

---

---

Que s'affiche-t-il quand on exécute le code :

```
→ for(int i(0); i < 3; ++i) {
  → for(int j(0); j < i; ++j) {
    cout << j;
  }
  cout << endl;
}
```

$i: 0 \ 1 \ 2$   
 $j: 0$

A:

0  
01

C:

rien

B:

0  
01  
012

D:

0123  
0123  
0123

allons détailler pourquoi. Comme avant, la première boucle for, initialise une variable i à zéro. La condition est "i strictement inférieure à 3", et l'instruction d'incréméntation est ++i. Donc, encore une fois, comme avant, i va prendre la valeur zéro, un et deux. On commence par entrer dans la boucle for i, avec i qui vaut zéro. Et on exécute, cette instruction-ci, qui est une deuxième boucle for, qui initialise une variable j à zéro.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

0m 5s



Que s'affiche-t-il quand on exécute le code :

```
→ for(int i(0); i < 3; ++i) {
    for(int j(0); j < i; ++j) {
        cout << j;
    }
    cout << endl;
}
```

$i: 0 \ 1 \ 2$   
 $j:$

↩

A:

0  
01

C:

rien

B:

0  
01  
012

D:

0123  
0123  
0123

Et la condition est  $j$  strictement inférieure à  $i$ .  $i$  vaut zéro.  $j$  vaut zéro également. Donc, cette condition est fausse. Et l'on sort, tout de suite de cette boucle for pour arriver ici Et arriver ici, ça veut simplement dire, qu'on fait un retour à la ligne, que je vais représenter, de cette façon-ci. Et on revient à cette instruction-ci,  $i$  va

notes

résumé

0m 49s



Que s'affiche-t-il quand on exécute le code :

```

-> for(int i(0); i < 3; ++i) {
    -> for(int j(0); j < i; ++j) {
        -> cout << j;
    }
    -> cout << endl;
}

```

$i: 0 \ 1 \ 2$   
↑

$j: 0$

↙  
0  
↘

A:

0  
01

C:

rien

B:

0  
01  
012

D:

0123  
0123  
0123

donc maintenant, prendre la valeur un. j va être initialisé à zéro. La condition est maintenant j strictement inférieur à i qui vaut un. Donc, j va prendre les valeurs, ou plus exactement, la valeur zéro. Et on va s'arrêter, tout de suite, dans la boucle. Donc, pour l'instant, j à la valeur zéro. On va exécuter cette instruction-ci. C'est-à-dire, afficher la valeur de j qui est zéro. On sort de la boucle, pour arriver à cette instruction, ici, qui fait un retour à la ligne. Et, on arrive à la fin de la boucle for i, et on va revenir,

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

.....

.....

.....

.....

.....

1m 25s



Que s'affiche-t-il quand on exécute le code :

```
→ for(int i(0); i < 3; ++i) {
    for(int j(0); j < i; ++j) {
        cout << j;
    }
    cout << endl;
}
```

$i: 0 \ 1 \ 2$   
 $j:$

↙  
 0 ↙

A:

0  
 01

C:

rien

B:

0  
 01  
 012

D:

0123  
 0123  
 0123

donc, ici, i va donc prendre la valeur deux.

notes

résumé

2m 13s



Que s'affiche-t-il quand on exécute le code :

```

→ for(int i(0); i < 3; ++i) {
  → for(int j(0); j < i; ++j) {
    → cout << j;
  }
  cout << endl;
}

```

Handwritten notes in red:

$i: 0 \ 1 \ 2$   
 $j: 0 \ 1$   
 (Arrows indicate the sequence of values for i and j)

A:  
 0  
 01

C:  
 rien

B:  
 0  
 01  
 012

D:  
 0123  
 0123  
 0123

On rentre dans cette boucle for i, pour exécuter cette instruction for, qui encore une fois, initialise j à zéro. La condition est maintenant "j strictement inférieur à deux". Donc, j va prendre la valeur zéro et un. On commence avec la valeur zéro dans j, et on exécute cette instruction cout.

notes

résumé

2m 17s





Que s'affiche-t-il quand on exécute le code :

```
→ for(int i(0); i < 3; ++i) {
  → for(int j(0); j < i; ++j) {
    → cout << j;
  }
  cout << endl;
}
```

A:

0  
01

B:

0  
01  
012

C:

rien

D:

0123  
0123  
0123

i: 0 1 2  
↑  
j: 0 1  
↑

↙  
0  
↘  
0

C'est-à-dire qu'on va afficher la valeur de j qui est zéro. On arrive à la fin de la boucle "for j", et on revient, ici, avec j qui va maintenant valoir un. On va donc exécuter cette instruction qui affiche la valeur de j, c'est-à-dire un, et on va sortir de la boucle. C'est-à-dire, exécuter cette instruction-ci, qui fait un retour à la ligne. On arrive à la fin de la boucle "for i". i vaut maintenant 3. Cette condition est fausse. Et on va sortir de la boucle pour arriver à la fin du code. Et on va donc bien afficher, le résultat qui était donné. qui était donné.

notes

résumé

2m 49s

