

Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Itérations - approfondissement et exemples (partie 4)

Concepts (extraits des sous-titres générés automatiquement) :

Boucles imbriquées. Boucle for. Boucle for i. Variable j. Condition j inférieur. Intérieur de ce branchement conditionnel. Opération d'incrémentation. Exemple de la table de multiplication. Nombre de notes. Corps de la boucle. Stade du programme. Variable i. Condition i inférieur. Branchement conditionnel. Première boucle for.



[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 : approfondissement et exemples

(Partie 4)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s



Une **solution**:

```
double note, somme(0.0);
int nombre_de_notes;

cout << "Entrez le nombre de notes" << endl;
cin >> nombre_de_notes;

if (nombre_de_notes > 0) {
    for(int i(1); i <= nombre_de_notes; ++i) {
        cout << "Entrez la note numero " << i << endl;
        cin >> note;
        somme = somme + note;
    }

    cout << "Moyenne = " << somme / nombre_de_notes << endl;
}
```

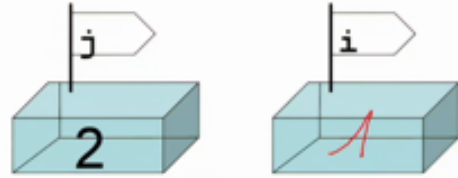
Cette erreur va survenir en fait quand l'utilisateur entre la valeur zéro pour la variable "nombre de notes". Et à ce stade du programme, je vais diviser par zéro, ce qui va provoquer une erreur. Alors, il y a plusieurs solutions possibles. Une solution, ça va être de tester si le nombre de notes est bien supérieur à zéro, comme il se doit, et donc je n'ai plus qu'à ajouter un branchement conditionnel, mettre le code qui demande d'entrer les notes à l'intérieur de ce branchement conditionnel, ainsi que

notes

résumé

0m 1s





```
for(int j(2); j <= 10; ++j) {
    cout << "Table de multiplication par " << j << ":" << endl;
    → for(int i(1); i <= 10; ++i) {
        cout << j << " multiplie par " << i << " vaut " << j * i << endl;
    }
}
```

Table de multiplication par 2:

1

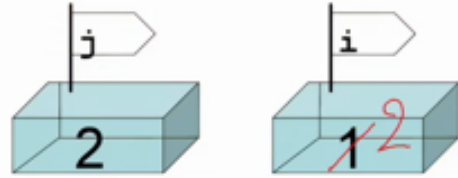
l'instruction qui affiche la moyenne des notes. Terminons avec ce qu'on appelle les boucles imbriquées. Alors je vais revenir sur l'exemple de la table de multiplication par 5 qui utilisait cette boucle for pour afficher la table, et je vais supposer maintenant que je veuille afficher toutes les tables de multiplication de 2 à 10. Et bien il suffit de mettre cette boucle for à l'intérieur d'une autre boucle for, et de la modifier légèrement. Alors si je mets directement cette boucle for à l'intérieur d'une autre, comme ce que j'ai fait ici, donc j'ai gardé exactement le même code pour cette boucle for, je l'ai mis à l'intérieur d'une deuxième boucle for qui utilise une variable j, qui va être initialisée à 2, avec une condition, j inférieur ou égal à 10. Donc j pourra prendre neuf valeurs, de 2 à 10. Donc tout ce code va m'afficher neuf fois la table de multiplication par 5. Si je veux maintenant afficher toutes les tables de multiplication de 2 à 10, il me suffit de modifier ce 5 pour le changer en la valeur de j, et surtout de modifier ce 5, ici. Donc je vais donc obtenir ce code-ci, que je vais détailler pour vérifier qu'il fonctionne correctement. La première boucle for commence par déclarer une variable j initialisée à 2. On va ensuite tester la condition j inférieur ou égal à 10, la condition est vraie, donc on va rentrer dans le corps de la boucle for j ; c'est-à-dire exécuter cette instruction-ci, qui affiche "table de multiplication par", ensuite la valeur de j, c'est-à-dire 2, ensuite un ":". On va maintenant exécuter la boucle for i qui se trouve ici, qui déclare donc, une variable i, et l'initialise à la valeur 1. On va ensuite tester la condition i inférieur ou égal à 10, qui est vraie.

notes

résumé

0m 49s





```
for(int j(2); j <= 10; ++j) {
    cout << "Table de multiplication par " << j << ":" << endl;
    → for(int i(1); i <= 10; ++i) {
        cout << j << " multiplie par " << i << " vaut " << j * i << endl;
    }
}
```

```
Table de multiplication par 2:
2 multiplie par 1 vaut 2
```

On va donc rentrer dans la boucle for i ; c'est-à-dire exécuter cette instruction-ci avec la valeur de j qui vaut 2. On va afficher multiplier par, ensuite la valeur de i qui vaut 1, ensuite "vaut", et ensuite la valeur de l'expression j fois i, qui vaut tout simplement 2. On va continuer ensuite à exécuter la boucle for i, c'est-à-dire qu'on va revenir ici. L'opération d'incrémenter va ajouter 1 à la valeur de i, donc i va prendre la valeur 2, la condition est toujours vraie,

notes

résumé

3m 13s

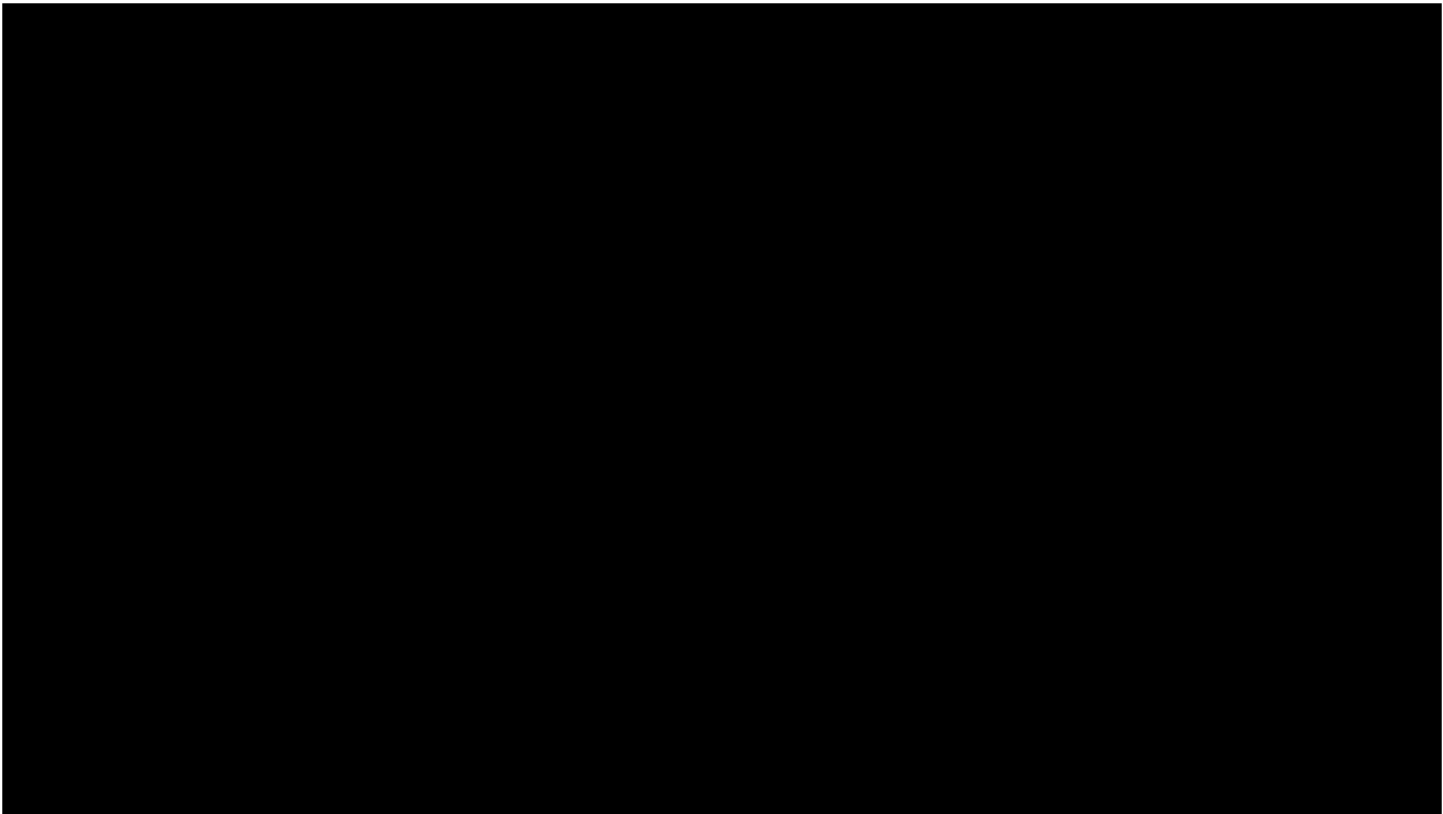


```
Table de multiplication par 2:
2 multiplie par 1 vaut 2
2 multiplie par 2 vaut 4
...
2 multiplie par 10 vaut 20
Table de multiplication par 3:
```

notes

4m 1s





Et on entre encore une fois dans la boucle for i. La variable i est de nouveau initialisée à 1. La condition est vraie, et on va entrer dans la boucle, dans le corps de la boucle for i, et exécuter cette instruction-ci, avec cette fois-ci j qui vaut 3, on va afficher "multiplié par". La valeur de i, qui vaut 1, ensuite "vaut", et la valeur de cette expression, qui est 3 fois 1, c'est-à-dire 3, et ainsi de suite. 1, c'est-à-dire 3, et ainsi de suite.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

5m 37s



.....

.....

.....

.....

.....