

Support de cours

Cours:

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

Vidéo:

### Itérations - approfondissement et exemples (partie 2)

Concepts (extraits des sous-titres générés automatiquement) :

**Variables différentes. Nouvel exemple. Boucles for. Avantage de cette nouvelle version. Nouvelle version. Stade du programme. Version équivalente. Boucle for. Transparent précédent. Intérieur d'une boucle for. Deuxième variable. Valeur de cette expression. Première valeur. Notes. Valeurs.**



[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 2)

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

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

0m 0s





Passons maintenant à un nouvel exemple qui est très

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

0m 1s



```
double note1;
cout << "Entrez la note numero 1" << endl;
cin >> note1;

double note2;
cout << "Entrez la note numero 2" << endl;
cin >> note2;

double note3;
cout << "Entrez la note numero 3" << endl;
cin >> note3;

double note4;
cout << "Entrez la note numero 4" << endl;
cin >> note4;

double somme(note1 + note2 + note3 + note4);
cout << "Moyenne = " << somme / 4 << endl;
```

notes

0m 5s



Sans boucle `for`, en n'utilisant que 2 variables:

```
double note, somme(0.0);

cout << "Entrez la note numero 1" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 2" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 3" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 4" << endl;
cin >> note;
somme = somme + note;

cout << "Moyenne = " << somme / 4 << endl;
```

pour stocker les quatre notes entrées par l'utilisateur, m'empêche d'introduire directement une boucle `for` pour réécrire ce programme. Donc, je vais commencer par modifier ce code en une version équivalente, qui n'utilise qu'une variable pour stocker les quatre notes entrées par l'utilisateur. Donc, voici cette nouvelle version, où je n'ai effectivement plus qu'une seule variable que j'ai appelée "note" qui va me servir à stocker les quatre notes entrées par l'utilisateur, et une deuxième variable,

notes

résumé

1m 1s



Pour vérifier le programme précédent, supposons que l'utilisateur entre les notes 5, 4, 6 et 4:

```
double note, somme(0.0);

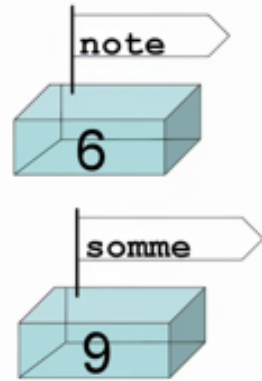
cout << "Entrez la note numero 1" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 2" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 3" << endl;
cin >> note;
somme = somme + note;

cout << "Entrez la note numero 4" << endl;
cin >> note;
somme = somme + note;

cout << "Moyenne = " << somme / 4 << endl;
```



la variable "somme", que j'ai initialisée ici à zéro. Donc, je vais demander à l'utilisateur d'entrer une valeur que je vais stocker dans ma variable "note", et j'ai ici une affectation qui s'écrit :  $\text{somme} = \text{somme} + \text{note}$ . Et je vais répéter quatre fois ce code. Et à ce stade du programme, la variable "somme" va contenir effectivement la somme des quatre valeurs entrées par l'utilisateur, et nous allons vérifier ceci sur un exemple. Pour vérifier cette nouvelle version, supposons que l'utilisateur entre les valeurs 5, 4, 6, et 4. La variable "note" n'est pas initialisée, donc on ne sait pas ce qu'elle contient pour l'instant, la variable "somme" a été initialisée à zéro. J'ai dit que la première valeur entrée par l'utilisateur, c'était 5, donc cette instruction-ci va mettre la valeur 5 dans la variable "note". On passe ensuite à cette affectation, qui va calculer la valeur de cette expression, c'est-à-dire, calculer  $\text{somme}$ , c'est-à-dire zéro, plus  $\text{note}$ , c'est-à-dire 5, obtenir la valeur 5 pour cette expression, et mettre cette valeur dans la variable "somme". La deuxième valeur entrée par l'utilisateur, ici, est la valeur 4, pour cette affectation. On va donc calculer maintenant  $\text{somme}$ , c'est-à-dire 5, plus  $\text{note}$ , c'est-à-dire 4, obtenir 9, et mettre ce 9 dans la variable "somme". Mais vous pouvez vous

notes

## résumé

1m 37s



Même programme en utilisant une boucle `for`.

```
double note, somme(0.0);

for(int i(1); i <= 4; ++i) {
    cout << "Entrez la note numero " << i << endl;
    cin >> note;
    somme = somme + note;
}

cout << "Moyenne = " << somme / 4 << endl;
```

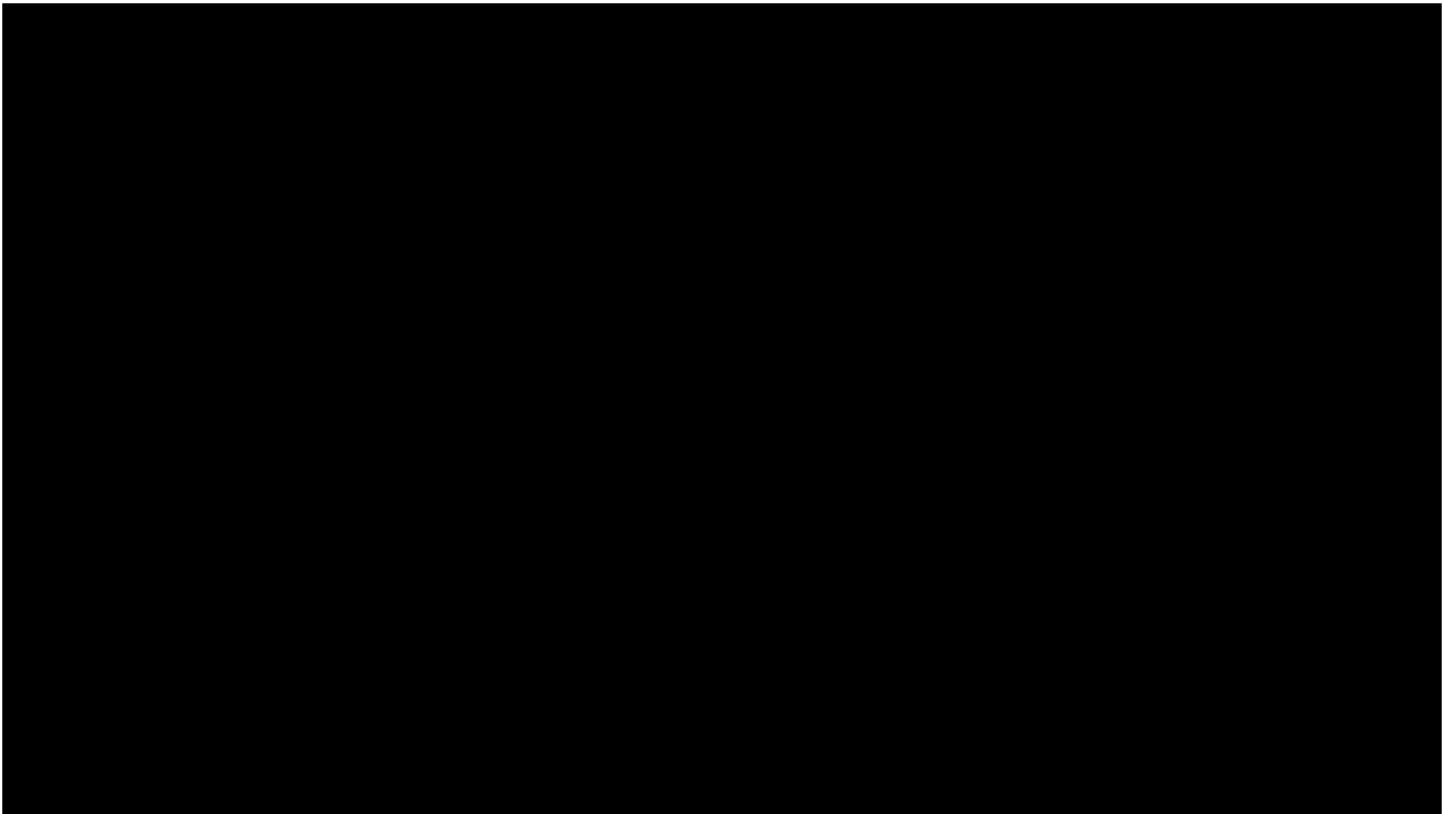
convaincre qu'à ce stade du programme, la variable "somme" va contenir zéro, plus 5, plus 4, plus 6, plus 4, c'est-à-dire bien la somme de quatre valeurs entrées par l'utilisateur. Alors l'avantage de cette nouvelle version, c'est que, on va pouvoir mettre ce code, qui est répété quatre fois, à l'intérieur d'une boucle `for`. C'est ce que j'ai fait ici, j'ai repris le code qui était sur le transparent précédent, j'ai juste modifié

notes

résumé

3m 25s





le numéro de la note, et utilisé une boucle for ici, avec une variable déclarée à l'intérieur de la boucle for, que j'ai appelée i, qui est initialisée à 1 et qui va aller jusqu'à 4. N'oubliez pas, dans cet exemple, d'initialiser la variable "somme" à zéro.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

résumé

4m 1s



.....