Omettre les délimiteurs

Regardons ce qui se passe si l'on écrit une boucle sans les accolades :

```
#include <iostream>
#define repeat(nb) for (int _loop = 1, _max = (nb); _loop <= _max; _loop++)
using namespace std;

int main()
{
   repeat (2)
      cout << "Bonjour !" << endl;
      cout << "Comment vas-tu ?" << endl;
}
```

↳

```
Bonjour !
Bonjour !
Comment vas-tu ?
```

Seule la première instruction d'affichage est répétée deux fois ! En effet, pour l'ordinateur, ce programme a le même sens que celui-là :

```
#include <iostream>
#define repeat(nb) for (int _loop = 1, _max = (nb); _loop <= _max; _loop++)
using namespace std;

int main()
{
   repeat (2)
   {
      cout << "Bonjour !" << endl;
   }
   cout << "Comment vas-tu ?" << endl;
}
```

Pour être plus précis, une boucle répète :

- soit les instructions d'un bloc entre accolades,
- soit une instruction terminée par un point-virgule « `;` ».

On peut donc omettre les délimiteurs quand on veut répéter une seule instruction. Toutefois, il faudra alors les rajouter dès qu'on ajoute une instruction, et si on oublie, cela peut être source d'erreurs. Beaucoup de développeurs informatiques mettent ainsi les accolades à chaque fois, notamment depuis l'étourderie notable dite Heartbleed. Nous vous suggérons donc de ne jamais les oublier !

Point-virgule mal placé

Observez aussi ce qu'il advient si l'on écrit une boucle en ajoutant par erreur un point-virgule à la fin de la première ligne, comme dans le programme ci-après.

```
#include <iostream>
#define repeat(nb) for (int _loop = 1, _max = (nb); _loop <= _max; _loop++)
using namespace std;

int main()
{
   repeat (2);
   {
```

```
        cout << "Bonjour !" << endl;
        cout << "Comment vas-tu ?" << endl;
    }
}
```

```
Bonjour !
Comment vas-tu ?
```

Les instructions ne sont pas répétées : en effet, la boucle ne répète que le point-virgule, qui est une instruction vide ! Après la boucle, le contenu du bloc qui suit est normalement exécuté, une fois. Prenez donc garde à ne pas mettre de point-virgule à cet endroit !