

Programmation : Comment demander de l'aide ?

Michel Billaud (michel.billaud@u-bordeaux.fr, michel.billaud@laposte.net)

3 juin 2020

Table des matières

1	Catégorie de problème	1
2	Problème de compilation	1
3	Fournir un code lisible	2
4	Description d'un problème d'exécution	2
5	Problème d'exécution : utilisez le compilateur	3
6	Scénario d'erreur minimal	3



Ce texte fait partie d'une petite collection de notes mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 2.0 France.

Vous travaillez sur un exercice ou un projet de programmation. Ça ne marche pas. Vous ne voyez pas pourquoi. Vous voudriez demander l'aide de quelqu'un. Bonne idée.

Il ne reste plus qu'à lui écrire. Mais, faute d'être extra-lucide, il/elle ne pourra vous aider qu'à partir des éléments que vous lui fournissez.

1 Catégorie de problème

Si vous débutez, vous affrontez de nombreux problèmes à la fois. C'est normal.

Donc commencez par préciser clairement si il s'agit

- d'un problème de **compilation** (syntaxe ou autre),
- d'un problème d'**exécution** (ça ne marche pas comme vous voulez),
- ou d'un problème de **conception** (vous ne voyez pas comment faire telle ou telle chose).

2 Problème de compilation

Vous avez tapé du code source, vous lancez la compilation, il y a un message (en rouge, d'habitude) qui dit qu'il y a un problème.

Pour avoir de l'aide à ce sujet, il faut que vous montriez au moins :

- le **message d'erreur complet** (celui de la première erreur).
- le **code source** en question, au minimum quelques lignes autour de celles qui posent problème.

Parfois, il n'est pas inutile d'indiquer aussi la **ligne de commande** que vous avez utilisée pour faire compiler le code source.

Attendez-vous à recevoir comme réponse "à votre avis, que veut dire le message que vous avez copié-collé?", qui signifie que

- l'erreur semble bien être causée par la ligne que vous indiquez,
- elle est triviale
- que vous auriez dû lire le message, qui vous décrit exactement le problème.

Exemple (en C)

1. Soit le code source `b.c`

```
#include <stdio.h>

int main()
{
    int pi = 3.14;
    printf("pi = %f\n", pi);
}
```

2. je le compile avec

```
$ gcc -Wall b.c
```

3. j'ai le message

```
b.c : In function 'main' :
b.c :6 :16 : warning : format '%f' expects argument of type 'double',
but argument 2 has type 'int' [-Wformat=]
    printf("pi = %f\n", pi);
                   ~^      ~~
                   %d
```

4. C'est la faute à Linux. Je vais réinstaller Windows.

3 Fournir un code lisible

Le code que vous fournissez doit toujours être **parfaitement indenté**, parce qu'une mauvaise indentation handicape la lecture, donc la recherche et la correction des erreurs.

- netbeans : shift-alt-F

- visual studio code : shift-control-I

4 Description d'un problème d'exécution

La compilation de votre code source s'est apparemment bien passée, mais "ça ne marche pas".

Ce n'est pas une description suffisante.

- on ne sait pas ce que votre programme est censé faire,
- il marche peut être dans certains cas, donc si vous avez constaté qu'il ne marche pas, donnez un **scénario d'erreur** avec
 - éventuellement les données que vous fournissez,
 - le résultat correct **attendu**,
 - le résultat incorrect que vous **constatez**.

Exemple

1. Je travaille sur le code C présenté plus haut
2. Je l'ai compilé avec la commande `gcc -o b b.c`
3. Je lance l'exécution
 - je m'attendais à voir `pi = 3.14`
 - en réalité je vois

```
$ ./a.out
pi = 0.000000
```

La réponse : sur cet exemple, on vous suggèrera de commencer par **recompiler le programme** avec les options qui produisent un maximum de messages d'avertissement.

N'oubliez pas que les enseignant-e-s sont là pour vous pousser vers une **démarche de travail correcte**, qui vous permettra d'être autonomes. Donc à moins d'être pressés de se débarrasser de vous et d'être sûr de ne plus jamais vous revoir - ils ne vont pas vous répondre directement "remplacez `int` par `double` ligne 6".

En fait, vos enseignants ont le *devoir* de se montrer si exaspérants à vous retourner des questions au lieu de répondre, que vous préféreriez vous les poser tout seul directement. C'est un métier sympa, en fait.

5 Problème d'exécution : utilisez le compilateur

Si vous aviez utilisé les options de compilation recommandées, le compilateur vous aurait signalé des problèmes dans votre code.

Donc :

- compilez **toujours** avec les options qui produisent le **maximum** de messages d'avertissement (En C ou C++, `-Wall -Wextra`, par exemple).
- tant que vous ne les avez pas **toutes** corrigées, inutile de lancer l'exécution.

6 Scénario d’erreur minimal

On a vu qu’il fallait fournir un **scénario d’erreur** pour décrire l’erreur.

Ce scénario d’erreur doit être le **plus petit possible**. Si votre programme qui afficher un tableau de nombres dans l’ordre croissant ne marche pas sur un exemple à 10 éléments, cherchez **activement** des exemples plus petits. Il est assez probable qu’on puisse trouver un tableau à 2 ou 3 éléments pour lequel ça ne marche pas (voire un seul!).

Intérêt double :

- on n’aura moins de doute sur le fait que le résultat soit faux.
- ça permettra de suivre plus facilement le déroulement de votre programme “à la main” ou avec un debugger.