

Langage C, TP 1

Juliusz Chroboczek

22 janvier 2026

Si vous utilisez Emacs, tapez

```
M-x customize-variable c-basic-offset
```

puis changez la valeur de la variable pour qu'elle vaille 4, puis validez. (La touche *Alt* s'appelle *Meta* sous Emacs, et *M-x* indique qu'il faut appuyer sur *Alt* en tapant *x*.)

Exercice 1 (Un premier programme C).

1. Créez un fichier `hello.c` dans lequel vous taperez le premier programme du premier poly. Compilez votre programme à l'aide de la commande

```
gcc -Wall bonjour.c
```

S'il y a des erreurs, corrigez-les et recommencez¹. Quel fichier a été créé? Exécutez-le à l'aide de la commande « `./a.out` ».

2. Supprimez le fichier `a.out`. Compilez votre programme de nouveau, mais cette fois-ci à l'aide de la commande

```
gcc -Wall -o bonjour bonjour.c
```

Quel fichier a été créé? Exécutez-le.

3. Si vous utilisez Emacs, tapez `M-x compile`. Remplacez la ligne de commande proposée par la ligne

```
gcc -Wall bonjour.c
```

puis validez. Si vous utilisez un autre environnement de développement (IDE), trouvez comment compiler depuis l'IDE.

Exercice 2. Écrivez un programme `nom.c` qui affiche votre nom. Compilez-le à l'aide de la commande « `gcc -Wall` », et testez-le.

Exercice 3. Écrivez un programme `agiste.c` qui demande à l'utilisateur son année de naissance puis affiche son âge. On supposera que ce programme ne sera utilisé qu'en 2026. Compilez et testez votre programme.

1. Vous n'êtes pas autorisés à consulter le chargé de TP avant d'avoir essayé de comprendre le message d'erreur sans aide.

Exercice 4. Écrivez un programme `pharmacie.c` qui demande à l'utilisateur le premier chiffre de son numéro de sécurité sociale puis répond « Bonjour Madame. » si l vaut 2, et « Bonjour Monsieur. » sinon. Compilez et testez votre programme.

Désormais, nous ne mentionnons plus que *tous les programmes que vous écrirez doivent être compilés et testés*.

Exercice 5. Écrivez un programme `max.c` qui affiche le plus grand de deux nombres entiers entrés par l'utilisateur.

Exercice 6. Modifiez le programme `pharmacie.c` pour qu'il affiche « Bonjour Madame. » si l'entrée était 2, « Bonjour Monsieur. » si l'entrée valait 1, et simplement « Bonjour. » sinon.

Exercice 7. Écrivez un programme `max3.c` qui affiche le plus grand de trois nombres entiers entrés par l'utilisateur. Vérifiez qu'il est correct même lorsque certains des nombres entrés sont égaux ((1, 1, 2), (1, 2, 1), (2, 2, 1), etc.).

Exercice 8. Écrivez un programme `un-deux.c` qui demande à l'utilisateur un entier positif puis affiche :

- « aucun » si ce nombre vaut 0;
- « un » si ce nombre vaut 1;
- « deux » si ce nombre vaut 2;
- « plusieurs » sinon.

Exercice 9. Écrivez un programme `bissextile.c` qui demande à l'utilisateur de rentrer le numéro d'une année puis indique s'il s'agit d'une année bissextile. Testez que votre programme est correct pour les années 2026, 2020, 1900, 2000.

Exercice 10. Écrivez un programme `heure.c` qui demande à l'utilisateur l'heure de la fin du TP (en heures et minutes, sous forme de deux entiers) et l'heure actuelle, puis affiche le nombre de minutes restant jusqu'à la fin du TP. N'utilisez que des variables entières.

Exercice 11. Dans l'ancien système monétaire britannique (le système « £sd »), une livre (*pound*, £) valait 20 sous (*shilling*, s), un sol valait 12 deniers (*pence*, d), un denier valait 4 *farthings* (f), et une guinée (*guinea*, gn) valait une livre et un sol.

1. Écrivez un programme qui prend en entrée deux sommes d'argent, chacune sous forme de trois entiers (livres, sous et deniers) et affiche leur somme, elle même sous forme de trois entiers. Attention, il faudra produire le résultat sous forme réduite : au plus 11 deniers et au plus 19 sous. Vous n'êtes pas autorisés à utiliser les nombres à virgule flottante.
2. En 1971, le gouvernement britannique a forcé la décimalisation du système : désormais, une livre vaut 100 *new pence*². Écrivez un programme qui lit une somme en livres et *new pence* (deux entiers) et affiche l'équivalent sous forme de 5 nombres : livres, sous, deniers, *farthings* et unités d'un douzième de *farthing*. Vous n'êtes toujours pas autorisés à utiliser les nombres à virgule flottante.

2. Les gens avaient peur que ce soit beaucoup trop compliqué.