

Tous les exercices en C seront écrits dans un fichier unique. Chaque exercice demande l'écriture d'une fonction et son utilisation dans la fonction `main()`. À chaque modification du contenu du fichier, il faut le sauver et compiler à nouveau

1. Écrire, sur papier, un bout de programme permettant d'échanger le contenu de deux variables `x` et `y`
2. Se connecter au serveur par `ssh` puis changer son mot de passe. Lancer l'éditeur de texte `anjuta` et ouvrir un nouveau fichier. Le sauver avec l'extension `.c`
3. Écrire une fonction `int demande(void)` qui demande à l'utilisateur un entier et le «retourne». Écrire une fonction `main()` utilisant cette fonction. Compiler le fichier dans le terminal et l'exécuter.
4. Écrire une fonction `float moyenne(int a, int b)` calculant le réel moyenne des entiers `a` et `b`. Cette fonction ne fera que calculer la moyenne sans contenir les commandes `scanf` ni `printf`. Pour utiliser cette fonction dans la fonction `main()` on utilisera deux fois la fonction `demande`
5. Écrire une fonction `resoudre` (à vous de trouver les paramètres, c'est à dire les ensembles de départ et d'arrivée), donnant la solution réelle d'une équation  $ax + b = 0$ . On supposera que le réel  $a$  est non nul
6. Écrire une fonction `float facto(int n)` «retournant»  $n!$
7. Écrire une fonction `float expo(int n)` «retournant»  $\sum_{k=0}^n \frac{1}{k!}$ . On utilisera la fonction `facto`
8. Écrire une fonction `float2 expo(int n)` «retournant»  $\sum_{k=0}^n \frac{1}{k!}$  sans utiliser la fonction `facto`. En quoi cette nouvelle fonction est plus économique que la précédente ?
9. Écrire une fonction «retournant» le terme d'indice  $n$  de la suite définie par  $u_0 = 1$  et pour tout entier  $n \geq 0$ ,  $u_{n+1} = \frac{u_n + 2}{u_n + 1}$
10. Écrire une fonction calculant le terme d'indice  $n$  de la suite de Fibonacci définie par  $u_0 = u_1 = 1$  et pour tout entier  $n \geq 2$ ,  $u_n = u_{n-1} + u_{n-2}$