

Classe inversée - Cours 2 et 3

Algorithmique et Programmation 1 - IMAC 1^e année

Cours 2 : compilation, variables, if, types simples, tableaux, chaînes de caractères

Cours 3 : itération, fonctions, paramètres, retour, pointeurs

Cours 2

Exercice 1. Compilation

- Créez un fichier `ping-pong.c` et avec l'aide d'un éditeur de texte (par ex. *emacs*) écrivez un programme qui écrit *pong*. Compilez. Exécutez.
- Faites une nouvelle version du fichier source (sauvegardez la nouvelle version sous un nouveau nom pour conserver l'historique de votre travail) :
 - déclarez un tableau de caractères,
 - insérez une instruction qui attend que l'utilisateur tape un mot au clavier et qui range le mot dans le tableau.Recompilez, exécutez, et tapez *ping*.

Exercice 2. Instruction conditionnelle

- Faites une nouvelle version de `ping-pong.c` de telle sorte que le programme n'écrive *pong* que si la 2^e lettre du mot tapé par l'utilisateur est *i*. Testez.
- Faites une dernière version dans laquelle le programme remplace la 2^e lettre par un *o*. Par exemple, si l'utilisateur écrit *bibine*, le programme doit répondre *bobine*. Testez.

Cours 3

Exercice 3. Fonctions

- Faites une nouvelle version de `ping-pong.c` dans laquelle tout le traitement précédent est fait par une nouvelle fonction, appelée par `main()`. Testez.
- Faites une nouvelle version qui répète 10 fois le traitement précédent.
- Faites une nouvelle version dans laquelle la répétition s'arrête avant la 10^e fois si l'utilisateur tape *fin*.
- Mettez les deux fonctions dans deux modules `.c` différents, faites un fichier d'en-tête `.h`, compilez et testez.

Exercice 4. Pointeurs

- Faites une nouvelle version de `ping-pong.c` dans laquelle la fonction qui contient l'itération compte le nombre de fois où l'utilisateur tape un mot. Ajoutez une instruction dans `main()` pour afficher ce résultat. Testez.
- Faites une nouvelle version dans laquelle on compte d'une part le nombre de fois où l'utilisateur tape un mot, et d'autre part le nombre de fois où la 2^e lettre du mot est un *i*. Testez.