

Algorithmique et Structures de données 1

LIV 2022-2023

Travaux Pratiques 1 - Rappels de C

Site du cours : <https://defelice.up8.site/algo-struct.html>

Les exercices marqués de (@) sont à faire dans un second temps.

Un fichier écrit en langage C se termine conventionnellement par `.c`.

Une commande de compilation est `gcc fichier_source1.c fichier_source2.c fichier_source3.c`.

Voici des options de cette commande.

- `-o nom_sortie` pour donner un nom au fichier de sortie (par défaut `a.out`).
- `-Wall` pour demander au compilateur d'afficher plus de Warnings
- `-Wextra` pour demander au compilateur d'afficher plus de Warnings
- `-std=c11` pour compiler selon la norme C11

Exemple : `gcc -Wall fichier1.c -o monprogramme`

Exercice 1. *Ech*

Écrire une procédure `void echange(int* a, int* b)` qui prend en paramètre les adresses de deux variables entières et qui échange les valeurs de ces variables.

Exemple d'utilisation :

```
int x=0;
int y=1;
echange(&x,&y);
// ici x==1 et y==0.
```

Exercice 2. *conversion tableau vers pointeurs*

Écrire une procédure `void afficher_tableau(int* T, int taille)` prenant en paramètre un tableau d'entiers `T` et sa taille `n` et qui affiche le contenu de ce tableau.

Exercice 3. *Random*

En utilisant la fonction `rand` de `<stdlib.h>` fabriquer une fonction `genererAlea(int a, int b)` qui génère une valeur entière aléatoire comprise entre `a` et `b`.

Exercice 4. *Le bonjour*

Écrire, dans un fichier `lebonjour.c`, un programme complet qui demande à l'utilisateur son nom et lui souhaite bonjour puis quitte.

Exemple :

```
./lebonjour
Quel est votre nom ?
Machin
Bonjour Machin. Au revoir
```

Exercice 5. *Le fichier*

Écrire, dans un fichier `lefichier.c`, un programme complet qui demande à l'utilisateur un nombre `n` puis enregistre dans un fichier `alea.nb`, `n` nombres aléatoires (en base dix et séparés par des espaces) tous compris entre 0 et 100.

Exercice 6. *bibliotheque*

Construire une bibliothèque statique contenant un fichier `maBibli.a` et un fichier `maBibli.h` qui contient deux fonctions.

1. `int somme(int n, int* T)` : qui fait la somme des nombres du tableau. Renvoie 0 si le tableau est vide.
2. `int produit(int n, int* T)` : qui fait le produit des nombres du tableau. Renvoie 1 si le tableau est vide.