

TP n°6 - Entrées et sorties

Dans un premier temps, créer un dossier **tp7** spécifique pour ce TP et copier les différents fichiers disponibles sur le répertoire partagé.

1 Échauffement

- **Q1.** Créer un fichier **fichier.txt** et un fichier C **tp6.c**.
- **Q2.** Dans **tp6.c**, ouvrir **fichier.txt** en écriture, y écrire votre nom avec **fprintf** puis le refermer. Compiler/exécuter le code. Vérifier le contenu du fichier.
- **Q3.** Modifier le code en mettant le nom de votre voisin au lieu du votre. Compiler/exécuter le code. Vérifier le contenu du fichier.
- **Q4.** Effacer le contenu du fichier.
- **Q5.** Modifier le code pour maintenant ouvrir en mode **ajout** et écrire votre nom. Compiler/exécuter le code. Vérifier le contenu du fichier.
- **Q6.** Enfin, remettre le nom de votre voisin au lieu du votre. Compiler/exécuter le code. Vérifier le contenu du fichier.

2 Une application : dessiner des graphes

Il n'y a pas de manière simple de tracer des courbes en C (à la manière de matplotlib en Python). À la place les programmeurs écrivent dans un fichier les coordonnées des points de la courbe (abscisse et ordonnée), et donnent le fichier en entrée à un logiciel comme **gnuplot**, qui fait l'affichage.

gnuplot n'est pas installé sur les ordinateurs du lycée. On va plutôt utiliser un petit programme Python **affiche.py** qui fait pareil (le programme est donnée dans le répertoire partagé).

- **Q7.** Écrire un programme **dessin_cube.c** qui crée un fichier **points_cube** (utiliser **fopen** sur un nom de fichier qui n'existe pas le crée) et le remplit avec les coordonnées de la fonction cube sur l'intervalle [0,100], pour des abscisses allant de 0,5 en 0,5.

Le contenu du fichier devrait être :

```
0 0
0.5 0.125
1 1
1.5 3.375
...
```

Il est important de respecter le nom du fichier et la mise en page des données (les espaces et les passages à la ligne). En revanche le nombre de chiffres après la virgule n'est pas important

- **Q8.** Dans le terminal, utiliser la commande suivante pour afficher : **python3 affiche.py**

3 Une application : copie de fichiers

- **Q9.** Écrire un programme **copy.c** qui prend en argument en ligne de commande deux noms de fichiers **nom1** et **nom2** et copie le contenu du fichier **nom1** dans le fichier **nom2**.
Après avoir compilé, on devrait pouvoir écrire dans le terminal **./copy.out nom1 nom2** et que ça fasse la copie.
- **Q10.** Créer deux fichiers, écrire quelques lignes dans l'un des deux et tester.

4 Une application : code César

Le but de cette partie est de décoder un message codé avec le code César.

Le principe du code de César est de choisir un nombre n entre 1 et 25 puis coder un texte en remplaçant chaque lettre par la n -ième suivante. Si il n'y a pas de n -ième lettre suivante, on retourne au début de l'alphabet.

Par exemple pour $n = 3$, en considérant uniquement l'alphabet français, le mot "zarbi" est encodé par "cduei". En effet, 3 lettres après le 'z', il y a le 'c', 3 lettres après le 'a' il y a le 'd', etc...

On vous donne un fichier **code.txt** d'un message secret codé en code César. Les caractères qui ne sont pas des lettres n'ont pas été changés.

Le n choisi est secret, mais on vous assure que le message originel est en français. Remarque : la lettre la plus commune en français est le 'e'.

- **Q11.** Écrire une fonction permettant de calculer n en se basant sur la fréquence des lettres.

- **Q12.** Décoder le texte et vérifier que le résultat fait sens. (Essayez de faire attention à la mise en page, les espaces et les passages de lignes)
Remarque : le 'a' a comme code ASCII 97, le 'b' est 98, etc... Pour obtenir le code d'un caractère, on peut demander à C de le considérer comme un `int`. Exemple : `int code = 'a';` donne la valeur 97 à `code`.

5 Une application : population de France

Dans cette section on met à votre disposition une base de données de la population des communes françaises : `population_communes.csv`

La base de données vient du site suivant et a été réduite pour éviter des problèmes de performances sur les ordinateurs du lycée : <https://public.opendatasoft.com/explore/dataset/population-francaise-communes/>.

Le fichier se présente ainsi : la première ligne contient le nom des champs. Chacune des lignes suivantes correspond à une commune. Il y a 4 champs : le nom de la région, le code du département (ex : 21 pour Dijon), le nom de la commune et sa population totale pour un certain recensement. **Les champs sont séparés par des points-virgules.**

Extrait du fichier :

```
Nom de la région;Code département;Nom de la commune;Population totale
Pays de la Loire;44;Nantes;290871
Occitanie;34;Montpellier;282143
Alsace;67;Strasbourg;276867
Nouvelle-Aquitaine;33;Bordeaux;256045
Hauts-de-France;59;Lille;238381
Nord-Pas-de-Calais;59;Lille;238381
Île-de-France;75;Paris 18e Arrondissement;198820
Île-de-France;75;Paris 19e Arrondissement;187939
...
```

Attention : il reste des doublons dans le fichier, par exemple Dijon est présent 2 fois : une fois avec la région Bourgogne (ancienne région) et une fois avec la région Bourgogne-Franche-Comté (nouvelle région). Il est donc normal que certaines réponses aux questions suivantes ne reflètent pas la réalité.

On pourra utiliser l'instruction C `strtok` qui permet de couper une chaîne de caractères en bouts selon un séparateur.

```
char* chaine = "Bonjour le monde";
char * bout = strtok (chaine, " ");

while (bout!=NULL){
    //On peut écrire des instructions ici
    //Puis on demande le token suivant
    bout = strtok(NULL, " ");
}
```

Dans ce code, `bout` vaudra successivement "bonjour", "le", "monde" et NULL.

- **Q13.** Premièrement, écrire un code qui détermine le nombre de lignes dans ce fichier. On rappelle que lire une ligne avec `scanf` alors qu'il n'en reste plus renvoie la valeur EOF, et qu'il suffit de tester la valeur renvoyée pour savoir si le fichier est fini.
- **Q14.** Afficher sur le terminal le nom des communes de plus de 100.000 habitants.
- **Q15.** Combien y a-t-il de communes dans le département de la côte d'or (21)? On fera attention au fait que les départements corses ont comme numéro 2A et 2B (donc pas des entiers).
- **Q16.** Combien de communes commencent par la lettre 'Q'? Par la lettre 'S'?
- **Q17.** Calculer la population totale de la France (métropole + départements d'outre-mer).
- **Q18.** Quel est le département le plus peuplé? Utiliser le code département pour se simplifier la tâche. On fera attention aux départements Corses (2A et 2B) ainsi qu'aux départements d'outre-mer : numéros 971, 972, 973, 974 et 976 (975 représentait St Pierre et Miquelon qui n'est plus un département).