

TP 12 : Entrées/Sorties sur fichiers

Imad Kissami

21 Avril 2025

Objectif

- Créer un `Makefile` pour compiler tous les fichiers `.cpp`.
- Tous les fichiers doivent être regroupés dans un dossier `TP12_Nom_Prénom`.
- Maîtriser les concepts d'entrées/sorties sur fichiers en C++ (lecture/écriture de fichiers, manipulateurs de flux, `stringstream`, validation des données).

Exercice 1 : Formatage de données dans un fichier

Écrivez un programme qui lit des nombres flottants à partir de l'entrée standard (terminée par une entrée vide) et les enregistre dans un fichier texte avec un formatage spécifique (précision de 2 décimales, justification à gauche, largeur de champ de 10). Affichez un message de confirmation une fois l'écriture terminée.

Instructions :

- Utilisez `std::cin` pour lire les nombres jusqu'à ce qu'une entrée non numérique soit détectée.
- Ouvrez un fichier `output.txt` en mode écriture avec `std::ofstream`.
- Utilisez les manipulateurs `std::fixed`, `std::setprecision`, `std::setw`, et `std::left` pour formater les nombres.
- Vérifiez que le fichier est correctement ouvert avant d'écrire.
- Affichez un message de succès ou d'erreur après l'opération.

Exemple du main.cpp :

```
int main() {
    std::ofstream out_file("output.txt");
    if (!out_file) {
        std::cerr << "Erreur_lors_de_l'ouverture_du_fichier" << std::endl;
        return 1;
    }

    double num;
    std::cout << "Entrez_des_nombres_flottants_(entrée_vide_pour_arrêter):_";
    while (std::cin >> num) {
        out_file << std::fixed << std::setprecision(2) << std::setw(10) << std::left << num << std::endl;
    }

    out_file.close();
    std::cout << "Nombres_enregistrés_dans_output.txt" << std::endl;
    return 0;
}
```

Exercice 2 : Analyse de fichier avec stringstream

Écrivez un programme qui lit un fichier `data.txt` contenant des lignes au format `nom age salaire` (par exemple, "Alice 30 50000.50"). Utilisez `std::stringstream` pour parser chaque ligne et générer un rapport formaté dans un fichier `report.txt` avec les colonnes alignées (nom, âge, salaire avec 2 décimales).

Instructions :

- Ouvrez `data.txt` avec `std::ifstream` et vérifiez son ouverture.
- Lisez chaque ligne avec `std::getline` et parsez-la avec `std::istringstream`.
- Écrivez le rapport dans `report.txt` avec `std::ofstream`, en utilisant `std::setw`, `std::left`, `std::fixed`, et `std::setprecision`.
- Vérifiez que les fichiers d'entrée et de sortie sont correctement ouverts.
- Affichez un message de succès ou d'erreur après l'opération.

Exemple du main.cpp :

```
int main() {
    std::ifstream in_file("data.txt");
    std::ofstream out_file("report.txt");

    if (!in_file || !out_file) {
        std::cerr << "Erreur_lors_de_l'ouverture_des_fichiers" << std::endl;
        return 1;
    }

    out_file << std::setw(20) << std::left << "Nom"
    << std::setw(10) << "Âge"
    << std::setw(15) << "Salaire" << std::endl;

    std::string line;
    while (std::getline(in_file, line)) {
        std::istringstream iss(line);
        std::string nom;
```

```
int age;
double salaire;
if (iss >> nom >> age >> salaire) {
    out_file << std::fixed << std::setprecision(2)
    << std::setw(20) << std::left << nom
    << std::setw(10) << age
    << std::setw(15) << salaire << std::endl;
}
}

in_file.close();
out_file.close();
std::cout << "Rapport_généré_dans_report.txt" << std::endl;
return 0;
}
```

Exercice 3 : Copie sélective de fichier avec validation

Écrivez un programme qui copie un fichier texte `input.txt` vers `output.txt`, en ne copiant que les lignes contenant un mot-clé saisi par l'utilisateur. Validez l'ouverture des fichiers et utilisez `std::getline` pour la lecture ligne par ligne.

Instructions :

- Demandez à l'utilisateur d'entrer un mot-clé via `std::cin`.
- Ouvrez `input.txt` avec `std::ifstream` et `output.txt` avec `std::ofstream`, en vérifiant leur ouverture.
- Lisez chaque ligne avec `std::getline` et vérifiez si elle contient le mot-clé avec `std::string::find`.
- Copiez uniquement les lignes contenant le mot-clé dans `output.txt`.
- Affichez un message de succès ou d'erreur après l'opération.

Exemple du `main.cpp` :

```
int main() {
    std::string keyword;
    std::cout << "Entrez le mot-clé à rechercher : ";
    std::getline(std::cin, keyword);

    std::ifstream in_file("input.txt");
    std::ofstream out_file("output.txt");

    if (!in_file || !out_file) {
        std::cerr << "Erreur lors de l'ouverture des fichiers" << std::endl;
        return 1;
    }

    std::string line;
    while (std::getline(in_file, line)) {
        if (line.find(keyword) != std::string::npos) {
            out_file << line << std::endl;
        }
    }

    in_file.close();
    out_file.close();
}
```

```
std::cout << "Lignes_\contentant_\\"" << keyword << "\"_\copiées_\dans_\output.txt" << std::endl;  
return 0;  
}
```