Projet SAE Shell

Ens.: Hanane Kteich, mail: kteich@cril.fr

10 septembre 2024

Table des matières

1 Introduction

L'objectif de ce projet est de réaliser un script Shell qui utilise un fichier de données d'entrée, plusieurs fonctions, et un fichier d'exécution principal (main.sh). Ce document décrit les exigences minimales pour réaliser ce projet, ainsi que les fichiers et fonctions nécessaires. Le but est de vous familiariser avec la programmation en Shell tout en structurant un projet professionnel.

2 Fichiers Requis

Le projet doit contenir plusieurs fichiers, chacun ayant un rôle spécifique.

2.1 Fichier de données d'entrée (`input.txt`)

Ce fichier contiendra des données variées telles que :
— Caractères (A, B, C, etc.)
— Chaînes de caractères (mots, phrases)
— Chiffres (1, 23, 456, etc.)
— Symboles (&, %, $, #)
— Séries de points et de tirets (ex : `...--...`, ajustable selon les besoins du projet)
L'organisation des données peut être adaptée au projet que vous choisissez de réaliser.

2.2 Fichier d'exécution principal (`main.sh`)

Ce fichier est le point d'entrée du projet. Il est utilisé pour exécuter l'ensemble du programme, lire les données d'entrée, et appeler les fonctions nécessaires. Il inclura les commandes pour importer les autres fichiers et lire les données du fichier d'entrée.

2.3 Fichier des fonctions (`functions.sh`)

Ce fichier (ou plusieurs fichiers selon la taille du projet) contiendra les fonctions définies pour accomplir les tâches du projet. Chaque fonction doit être bien documentée et chaque développeur doit ajouter son nom à côté de chaque fonction qu'il/elle a rédigée.

2.3.1 Règles à respecter

— Minimum de 6 fonctions par personne.
— Le fichier `main.sh` peut remplacer une fonction, sauf si ce fichier contient lui-même plusieurs fonctions, auquel cas chaque fonction devra être comptée individuellement.
— Au moins une fonction par personne doit accéder au fichier de données avec des paramètres spécifiques (et au moins une fonction doit lire une donnée à un endroit précis).
— Une fonction au moins doit utiliser la récursivité.
— Les fonctions simples comme `somme(a, b)` peuvent être utilisées dans d'autres fonctions pour éviter la répétition de code, mais elles ne comptent pas comme des fonctions principales.

2.4 Fichier de sortie (`output.txt`)

Ce fichier affichera les résultats des opérations effectuées sur les données d'entrée, ou les résultats peuvent être affichés directement dans le terminal.

3 Fonctions principales requises

Le projet doit inclure plusieurs fonctions principales avec des caractéristiques spécifiques. Voici les exigences pour ces fonctions :

- **Accès aux fichiers :** Une fonction doit être capable de lire des données spécifiques dans le fichier d'entrée, à des positions ou lignes données.
- **Récursivité :** Une fonction doit être écrite de manière récursive pour résoudre un problème donné (par exemple, calcul du factoriel d'un nombre, ou recherche récursive dans un fichier, etc.).
- **Fonctions utilitaires :** Des fonctions simples, comme additionner deux nombres, doivent être utilisées dans d'autres fonctions pour éviter de répéter des opérations.
- **Structure modulaire :** Le projet doit être organisé de manière modulaire, avec un découpage en plusieurs fichiers si nécessaire.

4 Conditions de livraison

4.1 Code source

Le code doit être bien commenté, avec le nom et le prénom de l'auteur de chaque fonction clairement mentionnés dans les commentaires.

4.2 Utilisation de Git

Utilisez Git pour versionner le projet. Cela vous permettra de suivre les modifications et de montrer le processus de développement du projet. (Pas obligatoire)

4.3 Rapport de projet

Le rapport doit inclure les éléments suivants :

- Contexte et description du projet.
- Choix des idées et des algorithmes utilisés.
- Explication des fichiers d'entrée et de sortie.
- Exemples d'exécution du programme.
- Documentation sur l'utilisation et les fonctionnalités du programme.

4.3.1 Format du rapport

Il est recommandé d'utiliser le format LaTeX pour le rapport, bien que ce ne soit pas obligatoire. Ce format vous permettra de vous habituer à la rédaction de documents professionnels.

5 Exemple de Structure de Projet

Voici un exemple de structure pour organiser votre projet Shell :

```
projet-shell/

 input.txt                  # Fichier d'entrée
 output.txt                 # Fichier de sortie (ou affichage terminal)
 main.sh                    # Fichier principal d'exécution
 functions.sh               # Fichier contenant les fonctions
 README.md                  # Documentation simple
 rapport.pdf (ou .doc)      # Rapport détaillé
```

6 Validation du Projet

Avant de commencer, venez valider votre idée de projet avec le professeur. Vous pouvez orienter votre projet vers un domaine professionnel qui vous intéresse (par exemple : automatisation des tâches, analyse de données, gestion de fichiers, cybre securité etc.).

7 Conclusion

Ce projet a pour objectif de vous initier à la programmation en Shell, tout en vous permettant de développer des compétences professionnelles en structurant un projet complet. Assurez-vous que votre code est clair, bien commenté, et qu'il respecte les exigences définies. N'oubliez pas de versionner votre travail, et d'inclure une documentation complète.

Bonne chance pour votre projet !