Grille d'évaluation pour le TP 2

Auteur du rendu évalué :

Nom du ou des fichiers évalués :

Évaluateur :

critère	Échec		Réussite	
	-	+	++	+++
Les programmes prévoient exhaustivement tous les cas d'utilisation.	Un cas d'utilisation normale fait échouer un programme.	Tous les cas d'utilisation normale sont traités correctement par les programmes.	Les programmes ne produisent pas d'erreur même en cas d'utilisation non conforme.	Les programmes informent l'utilisateur et le conseillent en cas de mauvaise utilisation.
Les identifiants de variables et fonctions décrivent fidèlement leur contenu.	Un élément du programme n'est pas correctement nommé et peut induire en erreur le relecteur.	Les éléments du programme sont correctement nommés.	Les identifiants suivent des règles de nomenclatures sur tout le code de manière cohérente.	Les identifiants apportent de l'information facilitant la compréhension par autrui.
Les programmes déploient des algorithmes raisonnables et efficaces.	Un programme utilise un mauvais algorithme ou procède à des calculs ou des tests inutiles.	Les programmes sont basés sur des algorithmes attendus.	Les programmes suppriment tout calcul non nécessaire à l'aide d'astuces simples.	Les programmes sont algorithmiquement plus performants que l'attendu (sans entrave aux consignes).
Les programmes sont correctement commentés.	Un passage trop rusé ou surprenant manque de commentaire ou encore un commentaire est présent mais non-éclairant.	Les programmes sont correctement commentés.	Des commentaires précisent les parties critiques du code, levant les ambiguïtés.	Les commentaires présentent les intentions et argumentent les choix de programmation de l'auteur.
Le programme répond correctement au cahier des charges (liste de fonctionnalités à coder).	Une des fonctionnalités n'est pas implémentée correctement.	Le programme répond correctement à toutes les tâches qu'il est censé effectuer.	Le programme contient quelques preuves de son bon fonctionnement.	Le programme démontre, durant son exécution, qu'il réalise correctement l'intégralité du cahier des charges.
Un makefile compile proprement toutes des sources automatiquement	Un des programmes n'est pas compilé.	Tous les programmes compilent produisant des exécutables fonctionnels.	Tous les programmes compilent sans erreur ni warning avec les flags -Wall -ansi.	Le makefile est bien lisible et gère correctement toutes les dépendances de ses sources.

Remarques :