Algorithmique et Structures de Données - Initiation

Rappel : Chaque projet comporte deux parties : la partie développement et la partie rapport. Même si certains projets peuvent être réalisés en binôme, ils seront soutenus individuellement et le rapport est donc propre à chaque étudiant. Pour la partie développement, quand aucun langage de programmation n'est spécifié, l'étudiant pourra choisir entre C ou JAVA. Concernant le rapport, il doit au minimum reprendre la problématique, la stratégie (la conception) envisagée pour la résoudre, le détail de la solution effective (+ si nécessaire, des extraits pertinents du code source) et enfin la conclusion et les références. Pour les projets réalisés en binôme, une section sur le travail en équipe est demandée.

Liste des projets

1. Calculatrice — (seul ou en binôme)

Écrire une calculatrice scientifique en ligne de commande (prendre pour exemple **bc**, saisir : `$ bc -l` dans un terminal). Cette calculatrice doit pouvoir parser toutes les expressions, gérer les opérations arithmétiques de base, le parenthésage, les priorités ainsi que les fonctions telles que : `sqrt, pow, cos, sin,` ... (voir les fonctions de `math.h`).

2. Rechercher - Remplacer — (seul)

Nous souhaitons développer un outil (une commande `shell`) facilitant les opérations de rechercher/remplacer dans des fichiers texte. Cet outil prend plusieurs entrées possibles :

- -f fichier1 [fichier2 [...]] : pour rechercher dans les fichiers cités ;
- find="mot1" : rechercher le pattern "mot1" ;
- replace="mot2" : remplacer par le pattern "mot2".

3. Agenda — (seul ou en binôme)

Nous souhaitons avoir un programme de gestion d'agendas. Nous pouvons Ajouter/Modifier/Supprimer des contacts. Chaque agenda est lu/enregistré à partir d'un fichier texte. Les contacts sont représentés sous la forme de listes chaînées. Nous avons pour chaque contact, les champs suivants : nom, prénom, adresse (num. de rue, rue, code postal, ville, pays), tél. dom., tél. bureau, fax, email — aucun doublon n'est permis. Tous ces champs peuvent servir à effectuer des recherches. Pour cela un pré-tri sur chaque champs doit être effectué (exemple : *Quick Sort* puis recherche dichotomique). Pour préserver la mémoire, chaque liste triée suivant un champs ne doit contenir que les pointeurs (ou références) vers les contacts.

4. Puissance 4 — (seul)

Créez un programme qui PERMET DE JOUER et JOUE au jeu Puissance4. On suppose que la largeur de la grille ainsi que le nombre de pièces devant être alignées pour gagner la partie sont définis au début de chaque partie par le joueur. Remarque : la hauteur de la grille est illimitée.

5. Calculs matriciels — (seul)

Écrire un programme qui, pour des matrices quelconques, effectue des additions, soustractions, multiplications et inversions matricielles.

6. Nombres premiers — (seul)

Écrire un programme C qui calcule (et affiche) la suite des nombres premiers s'arrêtant au plus grand premier représentable par le type entier lié à l'architecture.

7. π — (seul)

Écrire un programme qui calcule (et affiche) une approximation sur un `long double` des décimales du nombre π. Le programme devra converger vers la meilleure approximation et s'arrêter.

8. Jeu de Shannon — (seul)

Soit un graphe, dont deux noeuds (D et A) sont particuliers et deux joueurs, le lieur et le casseur. À chaque tour de jeu, le casseur casse une arête entre deux noeuds. À chaque tour de jeu, le lieur rend incassable une arête (non cassée) entre deux noeuds. Si, à un moment du jeu, il n'y a plus aucun moyen d'aller de D à A, le casseur a gagné. Si, à un moment donné, il existe un chemin incassable reliant D à A, le lieur a gagné.
Faites un programme qui permet à deux joueurs de jouer.
Faites un programme qui permet de jouer contre la machine.

9. Mots croisés — (seul)

Partant d'un fichier "dictionnaire" contenant des lignes "mot : définition", nous souhaitons écrire un programme qui génère une grille de mots croisés de dimensions quelconques (l'utilisateur spécifiera les dimensions dans la ligne d'arguments du programme). À chaque execution, la grille générée doit être (si possible) différente des précédentes.

10. Format BMP, PNG — (seul)

Écrire, en n'utilisant aucune API, une bibliothèque qui permet de lire et d'enregistrer des fichiers BMP et PNG. Pour tester cette bibliothèque, nous allons lire le fichier, dessiner sur l'image correspondante une diagonale et l'enregistrer dans un autre fichier.

11. C vs JAVA — (seul)

Pour un fichier contenant des entrées (au minimum 50000 lignes) alphanumériques classées dans un ordre quelconque, écrire et comparer une implémentation C et une implémentation JAVA d'algorithmes de tri (au minimum Insertion et Fusion).