

TD3 - Algorithmique et programmation

1/Manipulation de listes chaînées

- Définir les structures de données permettant de représenter une liste d'entiers.
- Écrire une fonction retournant un pointeur vers la dernière occurrence d'une valeur donnée dans une liste.
- Écrire un sous-programme supprimant toutes les occurrences d'une valeur de la liste.

2/Compression et décompression d'une image

Une image en 256 couleurs est codée par une matrice de pixels, chaque pixel prenant une valeur entre 0 et 255. On veut représenter une telle image par un tableau de listes chaînées, la i ème variable du tableau représente la $i+1$ ème ligne de l'image et contient un pointeur vers la liste des pixels de la ligne.

- Donner les types nécessaires à cette représentation.

Le stockage d'une image étant très onéreux en place mémoire, on voudrait compresser chaque ligne de la manière suivante : remplacer toute suite de n pixels identiques (i.e., de même couleur, et donc de même valeur v) par un couple (v, n) .

[illegible]

- Écrire un sous-programme permettant de compresser une ligne d'une image.
- Écrire un sous-programme permettant de compresser une image complète.
- Écrire les sous-programmes permettant de décompresser une ligne et une image.

3/Liste doublement chaînée

- Définir les structures de données nécessaires.
- Écrire une fonction qui supprime une valeur d'une liste.
- Écrire une fonction qui insère un nombre dans une liste déjà triée.

4/Vérification de parenthésage

Le but de cet exercice est d'écrire un petit programme qui lit au clavier une suite de caractères parmi {} () [] et vérifie que la position de ces caractères est cohérente. Pour cela, vous utiliserez une pile qui sera mise à jour après la saisie chaque caractère. Le caractère ! signifiera que la saisie est terminée.

5/Simulation

Donnez les déclarations ainsi que les opérations de base (empiler, dépiler, estvide) permettant de simuler une pile à l'aide de tableaux.

Même question pour la simulation d'une file.

6/Jeu de bataille

Rappel des règles

- nombre de joueurs : 2
 - chaque joueur dispose d'un tas de cartes, faces cachées
 - à chaque tour, les joueurs posent sur la table, face découverte, la carte placée sur le *dessus* de leur tas. Le joueur ayant découvert une carte de plus forte valeur ramasse les cartes placées sur la table et les met *sous* son tas. En cas d'égalité (bataille!), chacun pose une nouvelle carte sur la table (personne ne ramasse les cartes, elles sont remises en jeu pour le tour suivant).
- a) Définir les types de données nécessaires
 - b) Écrire une procédure qui, étant donné les deux tas de cartes des joueurs, joue à la bataille et affiche le gagnant.