

Travaux pratiques n°8: Fichiers texte: création, lecture et écriture + Allocation dynamique

Objectifs

- Comprendre la manipulation des fichiers en format texte en langage C.
- Manipuler l'allocation dynamique en langage C

Programmer la gestion des fichiers texte en langage C

Exercice 1: Lectures et écritures de chaînes

- a) Écrire un programme en C qui lit 10 chaînes tapées par l'utilisateur (d'une longueur maximale de 100 caractères) et qui les écrit dans un fichier dont le nom sera demandé à l'utilisateur.
- b) Écrire une procédure en C *lireChaine* qui ouvre un fichier texte en lecture et qui affiche à l'écran tous les enregistrements (d'une longueur maximale de 100 caractères) lus dans le fichier. Le nom du fichier est en paramètre. Faire un programme principal qui appelle cette procédure avec le fichier créé en a). **Rappel:** en fin de fichier, *fgetc* positionne un indicateur qu'on l'on teste avec la fonction *feof*.

Exercice 2: Lectures et écritures formatées

- a) Écrire une procédure en C *sauverEntText* qui sauvegarde les nombres de 100 à 110 (un par ligne) dans un fichier texte dont le nom est passé en paramètre. Faire un programme principal qui appelle cette fonction.
- b) Écrire une procédure en C *lireEnt* qui lit et affiche à l'écran les enregistrements d'un fichier texte dont le nom est en paramètre. Ces enregistrements sont des entiers. Faire un programme principal qui appelle cette procédure avec le fichier créé en a). **Rappel:** en fin de fichier, *fscanf* renvoie EOF et positionne un indicateur qu'on l'on teste avec la fonction *feof*.

Exercice 3:

Reprendre votre TP sur les structures et écrire une procédure de sauvegarde et de restauration d'un tableau de polygones.

Gérer l'espace dynamique en langage C

Exercice 4: Utilisation simple de malloc et free

1. Écrire un programme en C qui alloue dynamiquement la place pour un entier qui demande une valeur à l'utilisateur pour cet entier et qui l'affiche à l'écran avant de désallouer l'espace réserver pour cet entier.
2. Écrire un programme qui demande à l'utilisateur la taille d'un tableau d'entiers, qui réserve la place exacte pour ce tableau, qui demande les valeurs successives de ce tableau à l'utilisateur puis qui affiche le tableau et libère enfin la place prise par le tableau et qui répète ces opérations 5 fois.

Exercice 5: structures et allocation dynamique

Reprendre le TD sur les polygones. Dans cet exercice le tableau de points devra avoir la taille exacte du nombre de points qui constitue le polygone en utilisant l'allocation dynamique.

- a) Proposez une nouvelle structure *polygone*.
- b) Comment déclarer un tableau de polygones dont la taille sera donnée par l'utilisateur (donc allocation en espace dynamique).
- c) Écrire en C la fonction *creerPoly* qui permet de créer (**en allocation dynamique**) un nouveau polygone. La fonction a en paramètre le nombre de points du polygone. La fonction demande à l'utilisateur les points qui constituent ce nouveau polygone et retourne un pointeur sur ce polygone créé.
- d) Écrire en C un programme principal qui demande la taille du tableau de polygones à l'utilisateur et qui crée le tableau de polygones (en allocation dynamique) correspondant (en créant chacun des polygones grâce à la fonction *creerPoly* de c) .
- e) Écrire une procédure en C *supprPoly* qui reçoit un polygone de la forme définie en a) et qui désalloue la place occupée par ce polygone.