

Travaux dirigés n°13 (2h15):

Fichiers texte: création, lecture et écriture

Objectifs: Comprendre la manipulation des fichiers en format texte.

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, *fgets* 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: Lectures et écritures de structures. (à rendre pour la semaine prochaine)

- a) On vous propose une forme possible simple d'un fichier de sauvegarde en format texte d'un tableau de polygones (cf. TD sur les structures): en première ligne la taille du polygone suivi des points le constituant (un par ligne):

taillePoly1

x11,y11

x12,y12

...

taillePoly2

x21,y21

x22,y22

...

En utilisant ce format, donner en langage C, une procédure de sauvegarde (*sauvePolyText*), sous format texte, d'un tableau de polygones dans un fichier dont le nom est passé en paramètre.

b) Donner maintenant la procédure inverse *restaurPolyText* qui lit des polygones dans un fichier dont le nom est passé en paramètre et qui remplit un tableau de polygones passé en paramètre. On suppose que le nombre de polygone dans le fichier est inférieur ou égal à la taille du tableau, et idem pour les points le constituant.

c) Donner maintenant la procédure *sauvePolyTextBis* qui fait exactement la même chose que *sauvePolyText* mais pour un tableau **de pointeurs** de polygones.

Vous devez rendre:

Un programme commenté avec un jeu d'essais convaincant.
