Travaux Pratiques d'algorithmique n°5

Cours d'Informatique de Deuxième Année

—DEUG Sciences S3-MI—

Manipulation de fichiers, Tri

Ce TP permet de voir quelques opérations de base sur les fichiers en *C* ainsi qu'un algorithme de tri.

Pour manipuler un fichier en *C*, il faut utiliser une variable de type `FILE *` appelée un flux. Les fonctions suivantes permettent la lecture/écriture dans un fichier et le programme donné dans l'exercice 1 montre leur utilisation.

- `FILE *fopen(char *nom, char *mode);` : ouvre le fichier dont le nom est le premier paramètre selon le mode spécifié par le deuxième paramètre. Les principaux modes sont : `"r"` pour la lecture, `"w"` pour écriture et `"a"` pour l'ajout. Cette fonction retourne `NULL` en cas d'erreur ;
- `int fprintf(FILE *f,char *format,...);`
  et `int fscanf(FILE *f, char *format,...);` : ont le même comportement que les fonctions `printf` et `scanf` excepté le fait qu'elles lisent/écrivent dans le flux passé en premier argument. En particulier, `fscanf` retourne la constante particulière `EOF` (de type `int`) lorsque la fin du fichier est atteinte ;
- `int feof(FILE *f);` : retourne une valeur non nulle si la fin du flux est atteinte.
- `int fclose(FILE *f);` : ferme le flux `f`.

▸ **Exercice 1.** *Le programme suivant lit des entiers depuis un fichier et les recopie (après leur avoir ajouté 1) dans un autre fichier à raison d'un entier par ligne.*
Testez-le avec des fichiers que vous aurez créés à l'aide d'un éditeur.

```
#include <stdio.h>
int Copie(FILE *fp, FILE *gp)
{
  int x; int n;

  while ( (n = fscanf(fp,"%d",&x)) == 1 )
        fprintf(gp,"%d\n",x+1);
  if ( n == EOF )
     return 1;
  else return 0; /* une erreur de lecture s'est produite */
}

int CopieFichier(char *nomf, char *nomg)
{
```

```
  FILE * fp; FILE * gp; int res;

  fp = fopen(nomf,"r");
  if ( fp == NULL ) return 0;
  gp = fopen(nomg,"w");
  if ( gp == NULL ) return 0;
  res = Copie(fp,gp);
  fclose(fp); fclose(gp);
  return res;
}

int main(void)
{
  char nomf[21], nomg[21];

  printf("Donner le nom du fichier d'entree:");
  scanf("%20s",nomf);
  printf("Donner le nom du fichier de sortie:");
  scanf("%20s",nomg);
  if ( CopieFichier(nomf,nomg) == 0 )
     {
       fprintf(stderr,"ERREUR\n");
       return 1;
     }
  return 0;
}
```

Le test `if (n == EOF)` *dans la fonction* `Copie` *peut être remplacé par* `if ( feof(fp) )`.

▸ **Exercice 2.**
On va maintenant écrire et lire dans des fichiers les ensembles du TP précédent.

- *Écrire une fonction* `int EcritEnsemble(Ensemble e, char *nomfic)` *qui écrit l'ensemble* `e` *dans le fichier dont le nom est fourni. Que se passe-t-il si l'on utilise plusieurs fois cette fonction sur le même fichier ?*
- *Écrire une fonction* `int LitEnsemble(Ensemble *e, char *nomfic)` *qui construit l'ensemble* `e` *à partir des données du fichier dont le nom est fourni. Effectuer les tests nécessaires pour s'assurer que le fichier lu a le bon format, c'est-à-dire le même format que celui utilisé dans la fonction précédente.*

Ces fonctions retourneront 0 *en cas de problème,* 1 *sinon.*

▸ **Exercice 3.** ***(Tri à bulles)***
Le tri à bulle consiste à parcourir le tableau en permutant deux à deux les éléments qui ne sont pas dans l'ordre souhaité. Ainsi, lors du premier passage, on fera remonter (comme une bulle) l'élément le plus grand en haut du tableau. Le second parcours se fait donc sur un tableau de taille $n-1$ *et ainsi de suite.*
Écrire la fonction `TriABulle` *triant un ensemble. Évaluer sa complexité.*