

TD 4 – soutien – Fichier

Fichier texte

On souhaite traduire certains mots d'un texte à partir d'un lexique. Ce lexique est stocké dans un fichier texte dont chaque ligne est formatée comme suit : mot = traduction. Le texte à traduire est lui aussi sauvegardé dans un fichier texte. On créera un nouveau fichier texte qui sera la « traduction » du premier.

Exemple :

Soit le fichier de mots suivant :

```
si = if
alors = then
écrire = writeln
lire = readln
début = begin
fin = end
programme = program
entier = integer
```

Fichier texte donné

```
programme prog
var a : entier
début
  écrire('entrez un entier :')
  lire(a)
  si a > 2
    alors écrire('a est supérieur à 2')
fin
```

Fichier texte créé (traduction)

```
program prog
var a : integer
begin
  writeln('entrez un entier :')
  readln(a)
  if a > 2
    then writeln('a est supérieur à 2')
end
```

4.1. Quelle structure de donnée doit-on utiliser pour stocker le fichier de mots en mémoire principale ? Décrire son type.

```
Const max = 50
Type traduc = Enregistrement
              mot1, mot2 : chaine
              FinEnregistrement
tab-traduc = tableau [1..max] de traduc
```

4.2. On veut écrire la procédure charge-tableau qui permet de stocker le fichier de mots en mémoire principale. Expliquer en français le principe. Ecrire en pseudo-langage cette procédure.

```
Procédure charge-tableau(E nom : chaine ; S t : tab-traduc, i : entier)
Var f : FT
      c : chaine
Début
f←OuvreEnlecture(nom)
i←0
TantQue not Finfichier(f) Faire
  c←litchaine(f)
  i←i+1
  j←pos('=', c)
  t[i].mot1←copy(c,1,j-2)
  t[i].mot2←copy(c,j+2,lg(c)-j-1)
FinTantQue
Ferme(f)
Fin
```

4.3. Expliquer en français le principe de la traduction. Ecrire en pseudo-langage la procédure traduit-texte qui permet de créer le fichier texte de traduction à partir du fichier texte donné.

```
Fonction donne-traduc(m : chaine, t : tab-traduc, n : entier) : chaine
Var i : entier
      trouve : booléen
Début
trouve←faux
i←1
TantQue not trouve et i<=n Faire
  Si t[i].mot1=m
    Alors trouve←vrai
    Sinon i←i+1
  FinSi
FinTantQue
Si trouve=vrai
  alors retourner(t[i].mot2)
  sinon retourner('')
FinSi
Fin
```

```
Procédure traduit-texte(E nomfich, nomfichmot : chaîne)
Var ft, ftraduc : fichier texte
    nom, c, mot, mtraduc, ctraduc : chaîne
    t : tab-traduc
    nb,i : entier
Début
charge-tableau(nomfichmot ,t,nb)
ft←OuvreEnlecture(nomfich)
écrire(« entrer le nom du nouveau fichier : »)
lire(nom)
ftraduc←CréerFich(nom)
TantQue not Finfichier(ft) Faire
    c←litchaine(ft)
    ctraduc←c
    i←1
    TantQue i≤lg(c) Faire
        mot-suiv(c,i,mot)
        Si mot≠''
            Alors mtraduc←donne-traduc(mot,t,nb)
        Finsi
        Si mtraduc≠''
            Alors ctraduc←insère(efface(ctraduc,i-lg(mot),lg(mot)),mtraduc,i-lg(mot))
        Finsi
    FinTantQue
    ecritchaine(ftraduc,ctraduc)
FinTantQue
Ferme(ft)
Ferme(ftraduc)
Fin
```