

TD n° 2 et 3 d'Informatique

Gestion d'un carnet d'adresses

Le but de ces TD est de proposer une solution de gestion d'un ensemble d'informations concernant des personnes (nom, prénom, adresse, code postal, ville, téléphone, e-mail...)*. On désire pouvoir créer, consulter, supprimer et modifier ce répertoire.

- 1) Proposer une structure de données permettant de stocker les informations concernant une personne, puis une structure de données permettant de stocker durablement toutes les informations relatives à toutes les personnes.
- 2) Définir une procédure de saisie des données relatives à une personne, puis de stockage de ces données.
- 3) Définir une procédure permettant d'accéder à la liste des personnes et de l'afficher soit dans un fichier texte, soit à l'écran.
- 4) Définir une procédure de consultation des renseignements relatifs à une personne donnée. On suppose que l'on connaît la position de la personne dans le répertoire.
- 5) Définir une procédure de modification des renseignements relatifs à une personne donnée dont on connaît la position dans le répertoire.
- 6) Définir une procédure de suppression d'une personne donnée dont on connaît la position dans le répertoire.
- 7) Définir une procédure de recherche des renseignements concernant une personne à partir du nom et du prénom de la personne.
- 8) Définir une procédure permettant de lister toutes les personnes habitant une ville donnée.
- 9) Définir une procédure de tri par ordre alphabétique (nom + prénom) des personnes du répertoire.

Question subsidiaire : Une fois la liste triée, définir une procédure de recherche des renseignements concernant une personne plus rapide que celle de la question 7).

* Pour information, il faut savoir que la constitution et l'exploitation d'un fichier informatique contenant des informations sur des individus doit être soumis à l'accord de la CNIL (Commission Nationale Informatique et Liberté). Dans le cas contraire, de graves sanctions pénales sont encourues.

TP n° 2 et 3 d'Informatique

Gestion d'un carnet d'adresses

Le but de ces TP est d'implémenter en PASCAL les procédures conçues en TD. Écrire un programme principal avec menu permettant les opérations suivantes :

- 1) créer/recréer le fichier contenant les données ;
- 2) ajouter un ou plusieurs enregistrements ;
- 3) obtenir la liste des enregistrements à l'écran ou dans un fichier texte ;
- 4) afficher les renseignements concernant une personne à partir de sa position dans le fichier ;
- 5) modifier les données concernant une personne, sachant sa position dans le fichier ;
- 6) supprimer une personne du fichier, sachant sa position dans le fichier ;
- 7) rechercher les renseignements concernant une personne à partir de son nom et de son prénom ;
- 8) rechercher toutes les personnes habitant une ville donnée ;
- 9) trier le fichier par ordre alphabétique des noms et des prénoms des personnes.

À partir du fichier trié, on pourra également définir une procédure de recherche rapide des renseignements concernant une personne (à partir de son nom et de son prénom).

TP n° 2 et 3 d'Informatique

Gestion d'un carnet d'adresses : correction

```
{DEUG MIAS S2 - TP 2-3 : Gestion d'un carnet d'adresses}

program TP2_3;
uses crt;

{Declaration des types}

TYPE
  {Enregistrement}
  enr=record
    nom: string;
    prenom: string;
    adresse: string;
    cp: string;
    ville: string;
    tel: string;
    mail: string;
  end;
  {Fichier d'enregistrements}
  fic=file of enr;

{Declaration des variables globales}

VAR
  rep: fic;           {Fichier contenant le repertoire}
  choix: integer;

{Attente d'une touche au clavier}

PROCEDURE pause;
var c: char;
begin
  writeln;writeln('Presser une touche...');
  c:=readkey;
end;

{Creation du fichier de donnees f}

PROCEDURE creation(var f: fic);
var choix: integer;
begin
  writeln;
  repeat
    write('Detruire le fichier existant (1) ou Annuler (2) ? ');
    readln(choix);
  until (choix=1) or (choix=2);
  if choix=1 then
    begin
      rewrite(f);
    end;
end;
```

```

        close(f);
        writeln;writeln('Ok. ');
        pause;
    end;
end;

```

{Ajout d'enregistrements dans le fichier f}

```

PROCEDURE ajout(var f: fic);
var
    nouveau: enr;
    choix: integer;
begin
    {Positionnement en fin de fichier}
    reset(f);
    seek(f,filesize(f));
    repeat
        {Saisie}
        writeln;
        with nouveau do
            begin
                writeln('Numero      : ',filepos(f));
                write('Nom          : ');readln(nom);
                write('Prenom       : ');readln(prenom);
                write('Adresse      : ');readln(adresse);
                write('Code postal : ');readln(cp);
                write('Ville       : ');readln(ville);
                write('Telephone  : ');readln(tel);
                write('E-mail     : ');readln(mail);
            end;
        {Stockage}
        write(f,nouveau);
        {Stop ou encore}
        writeln;
        write('Saisir un nouveau correspondant (1) / Fin (0) ? ');readln(choix);
    until choix=0;
    close(f);
end;

```

{Liste des enregistrements du fichier f a l'ecran ou dans un fichier texte}

```

PROCEDURE liste(var f: fic);
var r: enr;
    res: text;
    choix, l: integer;
begin
    {Ecran ou texte ?}
    writeln;
    repeat
        write('Sortie a l''ecran (1) ou dans un fichier texte (2) ? ');
        readln(choix);
    until (choix=1) or (choix=2);
    writeln;
    if (choix=1) then assigncrt(res) {Fichier texte assigne a la console}
    else
        begin
            assign(res, 'rep.txt'); {Fichier texte assigne a rep.txt}
            writeln('Resultat dans le fichier REP.TXT. ');
        end;
    {Parcours de f}
    rewrite(res);
    reset(f);
    l:=0;
    while not eof(f) do

```

```

begin
  {Lecture d'un enregistrement dans f}
  read(f,r);
  l:=l+9; {L'affichage d'un enregistrement tient 8 lignes}
  if (choix=1) and (l>=24) then {Pause en bas de l'ecran}
    begin
      pause;
      l:=0;
    end;
  {Ecriture de l'enregistrement dans res}
  with r do
    begin
      writeln(res,' ');
      writeln(res,'Numero      : ',filepos(f)-1);
      writeln(res,'Nom        : ',nom);
      writeln(res,'Prenom     : ',prenom);
      writeln(res,'Adresse    : ',adresse);
      writeln(res,'Code postal : ',cp);
      writeln(res,'Ville      : ',ville);
      writeln(res,'Telephone  : ',tel);
      writeln(res,'E-mail     : ',mail);
    end;
  end;
  close(f);
  close(res);
  pause;
end;

{Saisie de position dans le fichier f}

FUNCTION saisiepos(var f: fic): integer;
var pos: integer;
begin
  writeln;
  reset(f);
  repeat
    write('Position dans le fichier (0-',filesize(f)-1,') : ');
    readln(pos);
  until (pos>=0) and (pos<=filesize(f)-1);
  close(f);
  saisiepos:=pos;
end;

{Affichage d'un enregistrement a partir de sa position dans le fichier f}

PROCEDURE acces(var f: fic);
var pos: integer;
    r: enr;
begin
  pos:=saisiepos(f);
  reset(f);
  seek(f,pos);
  read(f,r);
  with r do
    begin
      writeln;
      writeln('Numero      : ',pos);
      writeln('Nom        : ',nom);
      writeln('Prenom     : ',prenom);
      writeln('Adresse    : ',adresse);
      writeln('Code postal : ',cp);
      writeln('Ville      : ',ville);
      writeln('Telephone  : ',tel);
      writeln('E-mail     : ',mail);
    end;
  end;
end;

```

```

    end;
    close(f);
    pause;
end;

```

{Modification d'un enregistrement a partir de sa position dans le fichier f}

```

PROCEDURE modif(var f: fic);
var pos: integer;
    r: enr;
begin
    pos:=saisiepos(f);
    {Saisie}
    writeln;
    with r do
        begin
            writeln('Numero                : ',pos);
            write('Nouveau nom            : ');readln(nom);
            write('Nouveau prenom         : ');readln(prenom);
            write('Nouvelle adresse        : ');readln(adresse);
            write('Nouveau Code postal    : ');readln(cp);
            write('Nouvelle Ville             : ');readln(ville);
            write('Nouveau Telephone          : ');readln(tel);
            write('Nouvel E-mail                 : ');readln(mail);
        end;
    {Positionnement dans le fichier et stockage}
    reset(f);
    seek(f,pos);
    write(f,r);
    close(f);
end;

```

{Suppression d'un enregistrement a partir de sa position dans le fichier f}

```

PROCEDURE suppression(var f: fic);
var pos, i: integer;
    r: enr;
    temp: fic;
begin
    pos:=saisiepos(f);
    assign(temp, 'temp.fic');
    rewrite(temp);
    reset(f);
    {Copie du fichier f dans le fichier temp jusqu'a la position pos-1}
    for i:=0 to pos-1 do
        begin
            seek(f,i);
            read(f,r);
            write(temp,r);
        end;
    {Copie du fichier f dans le fichier temp de la position pos+1 a la fin}
    for i:=pos+1 to filesize(f)-1 do
        begin
            seek(f,i);
            read(f,r);
            write(temp,r);
        end;
    close(f);
    close(temp);
    {Suppression de l'ancien fichier f et remplacement par temp}
    erase(f);
    rename(temp, 'rep.fic');
    assign(f, 'rep.fic');
    writeln;writeln('Ok. ');
end;

```

```

    pause;
end;

{Recherche a partir du nom et du prenom dans le fichier f}

PROCEDURE recherche_nom(var f: fic);
var nc, pc: string;
    r: enr;
    trouve: boolean;
begin
    {Saisie nom a rechercher}
    writeln;
    write('Nom a chercher    : ');readln(nc);
    write('Prenom a chercher : ');readln(pc);
    trouve:=false;
    {Parcours sequentiel de f}
    reset(f);
    while (not eof(f)) and (not trouve) do
        begin
            read(f,r);
            if (r.nom=nc) and (r.prenom=pc) then trouve:=true;
        end;
    close(f);
    {Affichage resultat}
    writeln;
    if (trouve) then
        begin
            with r do
                begin
                    writeln('Adresse      : ',adresse);
                    writeln('Code postal : ',cp);
                    writeln('Ville      : ',ville);
                    writeln('Telephone  : ',tel);
                    writeln('E-mail     : ',mail);
                end;
            end
        else
            writeln('Personne ne correspond a ce nom et ce prenom...');
        pause;
    end;

{Recherche dans le fichier f des personnes habitant une ville donnee}

PROCEDURE recherche_ville(var f: fic);
var v: string;
    r: enr;
    c: integer;
begin
    writeln;
    write('Ville : ');readln(v);
    {Parcours de f avec affichage des habitants}
    writeln;
    reset(f);
    c:=0;
    while not eof(f) do
        begin
            read(f,r);
            if r.ville=v then
                begin
                    writeln(filepos(f),' ',r.nom,' ',r.prenom);
                    c:=c+1;
                end;
            end;
        end;
    close(f);

```

```

    if c=0 then writeln('Personne dans cette ville...');
    pause;
end;

```

{Tri du fichier f par ordre alphabetique (par permutation)}

```

PROCEDURE tri(var f: fic);
var r1, r2: enr;
    i, j: integer;
begin
    reset(f);
    for i:=0 to filesize(f)-2 do
        begin
            seek(f,i);
            read(f,r1);
            for j:=i+1 to filesize(f)-1 do
                begin
                    seek(f,j);
                    read(f,r2);
                    if (r1.nom+r1.prenom)>(r2.nom+r2.prenom) then
                        begin
                            {Permutation}
                            seek(f,i);
                            write(f,r2);
                            seek(f,j);
                            write(f,r1);
                            r1:=r2;
                        end;
                end;
            end;
        end;
    writeln;writeln('Ok. ');
    pause;
end;

```

{Recherche rapide a partir du nom et du prenom dans le fichier trie f}

```

PROCEDURE dichotomie(var f: fic);
var nc, pc: string;
    r: enr;
    trouve: boolean;
    min, max, pos, it: integer;
begin
    {Saisie nom a rechercher}
    writeln;
    write('Nom a chercher : ');readln(nc);
    write('Prenom a chercher : ');readln(pc);
    {Recherche dans f par dichotomie}
    reset(f);
    trouve:=false;
    it:=0;
    min:=0;
    max:=filesize(f)-1;
    while (min<=max) and (not trouve) do
        begin
            pos:=min+trunc((max-min)/2);
            seek(f,pos);
            read(f,r);
            it:=it+1;
            if (r.nom+r.prenom)=(nc+pc) then
                trouve:=true
            else if (r.nom+r.prenom)>(nc+pc) then
                max:=pos-1
            else
                min:=pos+1;
        end;
    end;

```



```

    end;
close(f);
{Affichage resultat}
writeln;
if (trouve) then
    begin
        with r do
            begin
                writeln('Adresse      : ',adresse);
                writeln('Code postal : ',cp);
                writeln('Ville      : ',ville);
                writeln('Telephone  : ',tel);
                writeln('E-mail     : ',mail);
                writeln;
                write('(trouve en ',it,' iteration)');
                if it>1 then write('s');
                writeln(')');
            end;
        end
    end
else
    writeln('Personne ne correspond a ce nom et ce prenom...');
pause;
end;

{Programme principal}

BEGIN
{Initialisations}
assign(rep,'rep.fic');
{Menu}
repeat
    clrscr;
    writeln;
    writeln('1) Creer/recreer le fichier de donnees');
    writeln('2) Ajouter des enregistrements');
    writeln('3) Liste des enregistrements');
    writeln('4) Affichage d'un enregistrement a partir de sa position');
    writeln('5) Modification d'un enregistrement a partir de sa position');
    writeln('6) Suppression d'un enregistrement a partir de sa position');
    writeln('7) Recherche a partir du nom et du prenom');
    writeln('8) Recherche des personnes habitant une ville');
    writeln('9) Tri par ordre alphabetique');
    writeln;
    writeln('666) Recherche rapide sur fichier trie');
    writeln;writeln('0) Fin');
    writeln;writeln;
    write('Votre choix : ');readln(choix);
    case choix of
        1: creation(rep);
        2: ajout(rep);
        3: liste(rep);
        4: acces(rep);
        5: modif(rep);
        6: suppression(rep);
        7: recherche_nom(rep);
        8: recherche_ville(rep);
        9: tri(rep);
        666: dicho(rep);
    end;
until choix=0;
END.

```