```
 67:
 68:
 69:
 70: listePersonne charger(int* nbrPersonne)
 71: {
 72:       int x;
 73:
 74:       listePersonne p = NULL;
 75:       listePersonne q = NULL;
 76:       ptrDataPersonne data = NULL;
 77:       listePersonne listeP = NULL;
 78:
 79:        //Ouverture du fichier listePersonne qui contient les donnees de chaque
    personne:
 80:       FILE* filePersonne = fopen("listePersonne.dta","rb");
 81:
 82:        //Verifier Si l'ouverture des fichiers est possible :
 83:       if( filePersonne == NULL )
 84:       {
 85:           printf("\n\n /!\\ Impossible d'ouvrir le fichier /!\\\n ");
 86:           getch();
 87:            //On quitte le programme :
 88:           exit(0);
 89:       }
 90:        // charger le nombre de personnes :
 91:       fread(nbrPersonne,sizeof(int),1,filePersonne);
 92:
 93:
 94:       // on charge la liste de personnes  :
 95:
 96:       // on crée un noeud personne :
 97:       p = Malloc(personne);
 98:
 99:       // listeP représente le début de la liste :
100:       listeP = p;
101:
102:       // on crée un noeud data dont lequel on charge les données
103:       // à partie du fichier listPersonne :
104:       data = Malloc(dataPersonne);
105:       p->data = data;
106:       p->suivant = NULL;
107:       fread(p->data,sizeof(dataPersonne),1,filePersonne);
108:
109:       q = p;
110:
111: do{
112:       q->suivant = p;
113:       q = p;
114:
115:        //On crée un noeud Personne et Data pour charger d'autre données
116:        // d'un autre personne :
117:       data = Malloc(dataPersonne);
118:       p = Malloc(personne);
119:       p->data = data;
120:       p->suivant = NULL;
121:
122:
123: }while(fread(p->data,sizeof(dataPersonne),1,filePersonne)!=0);
124:  //tant qu'on a encore des données dans le fichier listePersonne, on va les
    charger.
125:
126:        //Fermeture de fichier listePersonne :
```

```
127:        fclose(filePersonne);
128:
129:         //Ouverture de fichier relation :
130:        FILE* fileRelation = fopen("relation.dta","rb");
131:
132:        if( fileRelation == NULL )
133:        {
134:            printf("\n\n /!\\ Impossible d'ouvrir le fichier /!\\\n ");
135:            getch();
136:            exit(0);
137:        }
138:
139:         //On pointe sur le premier élément de la liste :
140:        p = listeP;
141:
142:         //On parcours la liste tant qu'on est pas encore arrivé à la fin :
143:        while( p != NULL )
144:        {
145:             fread(&x,sizeof(int),1,fileRelation);
146:             // si x=-1 donc la personne n'a pas de fils
147:             if( x == -1 ) p->fils = NULL;
148:             else
149:             // si non on appelle la fonction chercherPersonneById
150:             // qui va retourner l'adresse  du fils et le mettre
151:             // dans le noeud convenable
152:                  p->fils = chercherPersonneById(listeP,x);
153:
154:              // puis on va charger l id  du frére
155:             fread(&x,sizeof(int),1,fileRelation);
156:              // si x=-1 donc la personne n'a pas de frére
157:             if( x == -1 ) p->frere = NULL;
158:             else
159:             // si non on appelle la fonction chercherPersonneById
160:             // qui va retourner l'adresse  du frére et
161:             // le mettre dans le noeud convenable
162:                  p->frere = chercherPersonneById(listeP,x);
163:
164:             p = p->suivant;
165:        }
166:        // fermeture du fichier
167:        fclose(fileRelation);
168:         // on retourne le liste dans laquelle on a
169:         // charger  les données du fichier
170:        return listeP;
171: }
172:
173:
```