

```
1  #define MAX 100
2
3  struct NhanVien
4  {
5      char maNV[8];
6      char ho[10];
7      char tenDem[10];
8      char ten[8];
9      char diaChi[10];
10     int namSinh;
11     unsigned int luong;
12 };
13
14 typedef NhanVien Data;
15
16 struct NODE
17 {
18     Data info;
19     NODE* pNext;
20 };
21
22 struct LIST
23 {
24     NODE* pHead;
25     NODE* pTail;
26 };
27
28 NODE* GetNode(Data nv)
29 {
30     NODE* p = new NODE;
31     if (p != NULL)
32     {
33         p->info = nv;
34         p->pNext = NULL;
35     }
36     return p;
37 }
38
39 void CreateList(LIST& l)
40 {
41     l.pHead = l.pTail = NULL;
42 }
43
44 int IsEmpty(LIST l)
45 {
46     if (l.pHead == NULL)
47     {
48         return 1;
49     }
```

```
50     return 0;
51 }
52
53 void InsertTail(LIST& l, Data nv)
54 {
55     NODE* new_ele = GetNode(nv);
56     if (new_ele == NULL)
57     {
58         cout << "Loi!";
59         return;
60     }
61     if (l.pHead == NULL)
62     {
63         l.pHead = new_ele;
64         l.pTail = l.pHead;
65         return;
66     }
67     l.pTail->pNext = new_ele;
68     l.pTail = new_ele;
69 }
70
71 void InsertHead(LIST& l, Data nv)
72 {
73     NODE* new_ele = GetNode(nv);
74     if (new_ele == NULL)
75     {
76         cout << "Loi!";
77         return;
78     }
79     if (l.pHead == NULL)
80     {
81         l.pHead = new_ele;
82         l.pTail = l.pHead;
83         return;
84     }
85     new_ele->pNext = l.pHead;
86     l.pHead = new_ele;
87 }
88
89 void Nhap1NV(Data& nv)
90 {
91     cout << "\nNhap ma nhan vien: ";
92     cin >> nv.maNV;
93     cout << endl << "Nhap Ho: ";
94     cin >> nv.ho;
95     cout << endl << "Nhap ten dem: ";
96     cin >> nv.tenDem;
97     cout << endl << "Nhap ten: ";
98     cin >> nv.ten;
```

```
99     cout << endl << "Nhap dia chi: ";
100     cin >> nv.diaChi;
101     cout << endl << "Nhap nam sinh: ";
102     cin >> nv.namSinh;
103     cout << endl << "Nhap luong: ";
104     cin >> nv.luong;
105 }
106
107 void NhapThuCong(LIST& l)
108 {
109     Data nv;
110     int soNV;
111     CreateList(l);
112     cout << "\nNhap so nhan vien can them thong tin: ";
113     cin >> soNV;
114     for (int i = 0; i < soNV; i++)
115     {
116         Nhap1NV(nv);
117         InsertTail(l, nv);
118     }
119 }
120
121 void XuatTieuDe()
122 {
123     int i;
124     cout << endl;
125     //cout << "|";
126     for (i = 0; i < 70; i++)
127     {
128         cout << '=';
129     }
130     cout << "\n";
131     cout << setiosflags(ios::left);
132     cout << "|";
133     cout << setw(8) << "Ma SV"
134         << "|"
135         << setw(8) << "Ho"
136         << "|"
137         << setw(10) << "Ten lot"
138         << "|"
139         << setw(8) << "Ten"
140         << "|"
141         << setw(12) << "Dia chi"
142         << "|"
143         << setw(6) << "Nam Sinh"
144         << "|"
145         << setw(8) << "Luong"
146         << "|";
147 }
```

```
148     cout << endl;
149     //cout << "|";
150     for (i = 0; i < 70; i++)
151     {
152         cout << '=';
153     }
154 }
155 void Xuat1NV(Data nv)
156 {
157     cout << "\|";
158     cout << setiosflags(ios::left)
159         << setw(8) << nv.maNV << "|"
160         << setw(8) << nv.ho << "|"
161         << setw(10) << nv.tenDem << "|"
162         << setw(8) << nv.ten << "|"
163         << setw(12) << nv.diaChi << "|"
164         << setw(8) << nv.namSinh << "|"
165         << setw(8) << nv.luong << "|";
166 }
167 void XuatDS(LIST l)
168 {
169     XuatTieuDe();
170     for (NODE* p = l.pHead; p != NULL; p = p->pNext)
171     {
172         cout << "\n";
173         Xuat1NV(p->info);
174     }
175     cout << "\n";
176     for (int i = 0; i < 70; i++)
177     {
178         cout << '=';
179     }
180 }
181
182 int DocFile(char* filename, LIST& l)
183 {
184     Data nv;
185     ifstream in(filename);
186     if (!in)
187     {
188         return 0;
189     }
190     CreateList(l);
191     while (!in.eof())
192     {
193         in >> nv.maNV;
194         in >> nv.ho;
195         in >> nv.tenDem;
196         in >> nv.ten;
```

```
197         in >> nv.diaChi;
198         in >> nv.namSinh;
199         in >> nv.luong;
200         InsertTail(l, nv);
201     }
202     in.close();
203     return 1;
204 }
205
206 int DemSLChungHo(LIST& l, char* ho)
207 {
208     NODE* p = l.pHead;
209     int demHo = 0;
210     while (p != NULL)
211     {
212         if (_strcmphi(ho, p->info.ho) == 0)
213         {
214             demHo++;
215         }
216         p = p->pNext;
217     }
218     return demHo;
219 }
220
221 int DemSoNV(LIST& l)
222 {
223     int i = 0;
224     NODE* p = l.pHead;
225     while (p != NULL)
226     {
227         p = p->pNext;
228         i++;
229     }
230     return i;
231 }
232
233 double TBLuong(LIST& l)
234 {
235     double luongTB;
236     double tongLuong = 0;
237     NODE* p = l.pHead;
238     while (p != NULL)
239     {
240         tongLuong += p->info.luong;
241         p = p->pNext;
242     }
243     return tongLuong / DemSoNV(l);
244 }
245
```

```
246 int NamSinhTB(LIST& l)
247 {
248     double namTB;
249     double tongNam = 0;
250     NODE* p = l.pHead;
251     while (p != NULL)
252     {
253         tongNam += p->info.namSinh;
254         p = p->pNext;
255     }
256     return tongNam / DemSoNV(l);
257 }
258
259 void DaoNguocDS(LIST l)
260 {
261     LIST ll;
262     NODE* p = l.pHead;
263     if (IsEmpty(l))
264     {
265         cout << "\nDanh sach rong";
266         return;
267     }
268     CreateList(ll);
269     while (p != NULL)
270     {
271         InsertHead(ll, p->info);
272         p = p->pNext;
273     }
274     cout << "\nDanh sach sau khi dao nguoc:";
275     XuatDS(ll);
276 }
277
278 void HoanVi(Data& a, Data& b)
279 {
280     Data c = a;
281     a = b;
282     b = c;
283 }
284
285 void SXNoiBot(LIST& l)
286 {
287     NODE* p, * q;
288     for (p = l.pHead; p != NULL; p = p->pNext)
289     {
290         for (q = p->pNext; q != NULL; q = q->pNext)
291         {
292             if (q->info.namSinh < p->info.namSinh)
293             {
294                 HoanVi(q->info, p->info);
```

```
295     }
296     }
297 }
298 }
299
300 void XoaNVCoMSx(LIST& l, char x[8])
301 {
302     NODE* p = l.pHead;
303     if (_strcmpi(p->info.maNV, x) == 0)
304     {
305         l.pHead = p->pNext;
306     }
307     else
308     {
309         while (p != NULL)
310         {
311             if (_strcmpi(p->pNext->info.maNV, x) == 0)
312             {
313                 p->pNext = p->pNext->pNext;
314                 break;
315             }
316             p = p->pNext;
317         }
318     }
319 }
320
321 void ChenTrucTiep(LIST& l)
322 {
323
324     NODE* a, * b, * min;
325     for (a = l.pHead, a != NULL; a = a->pNext;)
326     {
327         min = a;
328         for (b = l.pHead; b != l.pTail; b = b->pNext)
329         {
330             if (b->info.luong > min->info.luong)
331                 HoanVi(b->info, min->info);
332         }
333     }
334 }
335 void DoiChoTrucTiep(LIST& l)
336 {
337     for (NODE* a = l.pHead; a != l.pTail; a = a->pNext)
338         for (NODE* b = a->pNext; b != NULL; b = b->pNext)
339         {
340             if (a->info.luong < b->info.luong)
341                 HoanVi(a->info, b->info);
342         }
343 }
```