

```

1 Listas SE - Recorrido:
2 Dada una lista simplemente enlazada de cadenas, retornar la cantidad
3 que tienen longitud par
4
5 typedef struct nodo {
6     char dato[15];
7     struct nodo * sig; } nodo;
8 typedef struct nodo * TLista;
9
10 int CantPares (TLista L){
11     TLista aux;
12     int cont = 0;
13     aux = L;
14     while (aux != NULL) {
15         if (!(strlen(aux-> dato) % 2))
16             cont++;
17         aux = aux-> sig;
18     }
19     return cont;
20 }
21
22 -----
23 Listas SE - Búsqueda:
24 Dada una lista simplemente enlazada de cadenas, verificar si X está
25
26 int esta (TLista L, char *x){
27     TLista aux;
28     aux = L;
29     while (aux != NULL && strcmp(x, aux-> dato) != 0)
30         aux = aux-> sig;
31     return aux != NULL;
32 }
33
34 -----
35 Dada una lista simplemente enlazada ordenada de cadenas, verificar si
36 X está
37
38 int estaOrd (TListaE L, char *x){
39     TListaE aux;
40     aux = L;
41     while (aux != NULL && strcmp(x, aux-> dato) > 0)
42         aux = aux-> sig;
43     return aux != NULL && strcmp(x, aux-> dato) == 0;
44 }
45
46 -----
47 Listas SE - Inserción
48 Insertar un dato en una lista ordenada simplemente enlazada de
49 enteros
50
51 void insertaOrd (TListaE * L, int x){
52     TListaE aux, ant, act;
53     aux = (TListaE) malloc (sizeof(nodo));
54     aux-> dato = x;
55     if (*L == NULL || x < (*L)-> dato) {
56         aux-> sig = *L;
57         *L = aux;
58     }
59     else{
60         ant=NULL;
61         act=*L;
62         while (act != NULL && x > act-> dato){
63             ant=act;
64             act=act-> sig;
65         }
66         aux-> sig=act;

```

```

67     ant->sig=aux;
68 }
69 }
70
71 -----
72 Listas SE - Eliminación
73 Dada una lista ordenada simplemente enlazada de enteros, eliminar X
74
75 void eliminaE(TListaE *L, int x){
76     TListaE ant,act;
77     if (*L !=NULL)
78         if ((*L)->dato==x){
79             act=*L;
80             *L=(*L)->sig;
81             free(act);
82         }
83     else {
84         act = *L;
85         while (act != NULL && x>act->dato){
86             ant=act;
87             act = act -> sig;
88         }
89         if (act != NULL && x==act->dato) {
90             ant->sig=act->sig;
91             free(act);
92         }
93     }
94 }

```