

```

1: program DLstEnc;
2: uses LstEnc;
3:
4: var
5:     Lista : Lista_Encadeada;
6:     Item : Tipo_Item;
7:     Ok : boolean;
8:
9: procedure ImprimirConteudo(var Lista : Lista_Encadeada);
10: var P : Pont_Item_Lista;
11: begin
12:     P := Lista.Cabeca;
13:
14:     while P <> nil do
15:         with P^.Item do
16:             begin
17:                 writeln(Chave, ' - ', Dado);
18:                 P := P^.Proximo
19:             end
20: end;
21:
22: BEGIN
23:     Inicializar(Lista);
24:
25:     Item.Chave := 1;
26:     Item.Dado := 'A';
27:     if Inserir(Item, Lista) then
28:         writeln('Inseriu 1');
29:
30:     Item.Chave := 2;
31:     Item.Dado := 'B';
32:     Inserir(Item, Lista);
33:
34:     writeln('Tamanho:', Tamanho(Lista));
35:     writeln('Cheia:', Cheia(Lista));
36:     writeln('Vazia:', Vazia(Lista));
37:
38:     ImprimirConteudo(Lista);
39:
40:     Item.Dado := 'C';
41:     if Alterar(Item, Lista) then
42:         writeln('Alterou 2');
43:
44:     ImprimirConteudo(Lista);
45:
46:     Obter(2, Lista, Item, Ok);
47:     if Ok then
48:         writeln('Chave:', Item.Chave, ' Dado:', Item.Dado);
49:
50:     if Remover(2, Lista) then
51:         writeln('Removeu - Tamanho = ', Tamanho(Lista));
52:
53:     ImprimirConteudo(Lista);
54:
55:     Apagar(Lista);
56:     writeln('Vazia:', Vazia(Lista));
57: END.

```