

```

1: program DLstCirC;
2: uses LstCirC;
3:
4: var
5:     Lista : Lista_Circular;
6:     Item : Tipo_Item;
7:     Ok : boolean;
8:
9: procedure ImprimirEstado(var Lista : Lista_Circular);
10: begin
11:     writeln('Tamanho:', Tamanho(Lista));
12:     writeln('Vazia:', Vazia(Lista));
13:     writeln('Cheia:', Cheia(Lista));
14: end;
15:
16: procedure ImprimirConteudo(var Lista : Lista_Circular);
17: var
18:     P : Pont_Item_Lista;
19: begin
20:     P := Lista.Cabeca^.Proximo;
21:
22:     while P <> Lista.Cabeca do
23:         with P^.Item do
24:             begin
25:                 writeln(Chave, ' - ', Dado);
26:                 P := P^.Proximo
27:             end
28: end;
29:
30: BEGIN
31:     Inicializar(Lista);
32:
33:     Item.Chave := 1;
34:     Item.Dado := 'A';
35:     if Inserir(Item, Lista) then
36:         writeln('Inseriu 1');
37:     ImprimirEstado(Lista);
38:
39:     Item.Chave := 2;
40:     Item.Dado := 'B';
41:     Inserir(Item, Lista);
42:
43:     ImprimirEstado(Lista);
44:     ImprimirConteudo(Lista);
45:
46:     Item.Dado := 'C';
47:     if Alterar(Item, Lista) then
48:         writeln('Alterou 2');
49:
50:     ImprimirConteudo(Lista);
51:
52:     Obter(2, Lista, Item, Ok);
53:     if Ok then
54:         writeln('Chave:', Item.Chave, ' Dado:', Item.Dado);
55:
56:     if Remover(2, Lista) then
57:         writeln('Removeu 2');
58:
59:     ImprimirEstado(Lista);
60:     ImprimirConteudo(Lista);
61:

```

```
62:    Apagar(Lista);
63:    writeln('Vazia:', Vazia(Lista));
64: END.
```