

```

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

```

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