

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  const int MAX_LIVROS = 100; // constante que determina o número máximo de livros que serão armazenados
7
8  // estrutura que define as propriedades do livro
9  struct Livro
10 {
11     int codigo; // código do livro
12     string nome; // nome do livro
13 };
14
15 // estrutura que define as propriedades da lista de livros
16 struct ListaDeLivros
17 {
18     int quantidade = 0; // armazena quantos livros estão cadastrados
19     Livro livros[MAX_LIVROS]; // vetor do tipo Livro para armazenar os livros na lista
20 };
21
22 ListaDeLivros lista; // determina uma variável do tipo listaDeLivros

```

```

24 // chama as funções no início do código para organização
25 void inserirInicio(Livro li);
26 void inserirFim(Livro li);
27 void inserirEmUmaPosicao(Livro li, int posicao);
28 void removerInicio();
29 void removerFim();
30 void removerDeUmaPosicao(int posicao);
31 void mostrarLista();
32 Livro leDoTeclado();
33 int menu();

```

```

35 // função principal onde temos um menu para que o usuário escolha as opções e assim o programa flua
36 int main()
37 {
38     int opcao, posicao; // variáveis responsáveis por obter as opções do menu e a posição do livro na lista
39     Livro livro; // determina um tipo de livro
40
41     // executa o menu enquanto a opção for diferente
42     do
43     {
44         opcao = menu(); // executa a função menu recebendo o valor da variável opção
45
46         // executa o menu com base na opção
47         switch (opcao)
48         {
49             case 1:
50                 livro = leDoTeclado(); // executa a função para ler as propriedades do livro pelo teclado
51                 inserirInicio(livro); //
52                 break;
53             case 2:
54                 livro = leDoTeclado(); // executa a função para ler as propriedades do livro pelo teclado
55                 cout << "Em qual posicao? ";
56                 cin >> posicao; // le a posição
57                 inserirEmUmaPosicao(livro, posicao); // executa a função
58                 break;
59             case 3:
60                 livro = leDoTeclado(); // recebe o livro
61                 inserirFim(livro); // insere no fim o livro cadastrado
62                 break;
63             case 4:
64                 removerInicio(); // remove o livro do início
65                 break;

```

```

66         case 5:
67             cout << "Em qual posicao? ";
68             cin >> posicao; // lê a posição
69             removerDeUmaPosicao(posicao); // remove o livro da posição
70             break;
71         case 6:
72             removerFim(); // remove o livro do fim
73             break;
74         case 7:
75             mostrarLista(); // mostra a lista de livros
76             break;
77         case 0:
78             cout << "Saindo... " << endl; // sai do menu
79             break;
80         default:
81             cout << "Opção inválida!" << endl; // se a opção for inválida
82             break;
83     }
84 } while (opcao != 0);
85 }

```

```

87 // função para inserir um novo livro na primeira posição recebendo li (estrutura Livro)
88 void inserirInicio(Livro li)
89 {
90     if (lista.quantidade < MAX_LIVROS) // se a quantidade de livros for menor que o máximo de livros permitidos ele:
91     {
92         for (int i = lista.quantidade; i > 0; i--) // percorre a lista de trás para frente
93         {
94             lista.livros[i] = lista.livros[i - 1]; // passa todos os livros uma posição para frente deixando a primeira posição vazia
95         }
96         lista.livros[0] = li; // atribui o novo livro à primeira posição
97         lista.quantidade = lista.quantidade + 1; // adiciona um livro na contagem de livros
98     }
99 }

```

```

101 // função para inserir um novo livro na primeira posição
102 void inserirFim(Livro li)
103 {
104     if (lista.quantidade < MAX_LIVROS) // se a quantidade de livros for menor que o máximo de livros permitidos ele:
105     {
106         lista.livros[lista.quantidade] = li; // insere o novo livro na última posição (a quantidade de livros já indica a última)
107         lista.quantidade = lista.quantidade + 1; // adiciona um livro na contagem de livros
108     }
109 }
110
111 // função para inserir um novo livro em uma determinada posição
112 void inserirEmUmaPosicao(Livro li, int posicao)
113 {
114     if (lista.quantidade < MAX_LIVROS) // se a quantidade de livros for menor que o máximo de livros permitidos ele:
115     {
116         for (int i = lista.quantidade; i > posicao; i--) // percorre a lista de trás para frente até encontrar a posição
117         {
118             lista.livros[i] = lista.livros[i - 1]; // passa todos os livros uma posição para frente
119         }
120         lista.livros[posicao] = li; // atribui o novo livro na posição
121         lista.quantidade = lista.quantidade + 1; // adiciona um livro na contagem de livros
122     }
123 }

```

```

125 // função para remover o primeiro livro da lista
126 void removerInicio()
127 {
128     if (lista.quantidade > 0) // se a quantidade de livros for maior que zero:
129     {
130         cout << "Removendo ";
131         cout << lista.livros[0].codigo << ' '; // imprime o código do livro que vai ser excluído
132         cout << lista.livros[0].nome << endl; // imprime o nome do livro que vai ser excluído
133         for (int i = 0; i < lista.quantidade; i++) // percorre a lista em ordem crescente
134         {
135             lista.livros[i] = lista.livros[i + 1]; // passa todos os livros uma posição para trás
136         }
137         lista.quantidade = lista.quantidade - 1; // remove um livro na contagem de livros
138     }
139 }
140
141 // função para remover o último livro da lista
142 void removerFim()
143 {
144     if (lista.quantidade > 0) // se a quantidade de livros for maior que zero:
145     {
146         cout << "Removendo ";
147         cout << lista.livros[lista.quantidade - 1].codigo << ' '; // imprime o código do livro que vai ser excluído
148         cout << lista.livros[lista.quantidade - 1].nome << endl; // imprime o nome do livro que vai ser excluído
149         lista.quantidade = lista.quantidade - 1; // remove um livro na contagem de livros
150     }
151 }

```

```

153 // função para remover o livro de uma determinada posição da lista
154 void removerDeUmaPosicao(int posicao)
155 {
156     if (lista.quantidade > 0 && posicao < lista.quantidade) // se a quantidade for maior que zero e a posição for menor que o valor da
        quantidade de livros cadastrados
157     {
158         cout << "Removendo ";
159         cout << lista.livros[posicao].codigo << ' '; // imprime o código do livro que vai ser excluído
160         cout << lista.livros[posicao].nome << endl; // imprime o nome do livro que vai ser excluído
161         for (int i = posicao; i < lista.quantidade; i++) // percorre a lista em ordem crescente
162         {
163             lista.livros[i] = lista.livros[i + 1]; // passa todos os livros uma posição para trás
164         }
165         lista.quantidade = lista.quantidade - 1; // remove um livro na contagem de livros
166     }
167 }
168
169 // função para imprimir a lista de livros
170 void mostrarLista()
171 {
172     cout << endl;
173     cout << "++++++ CONTEÚDO DA LISTA ++++++ " << endl;
174     for (int i = 0; i < lista.quantidade; i++)
175     {
176         cout << lista.livros[i].codigo << ' ';
177         cout << lista.livros[i].nome << endl;
178     }
179     cout << endl;
180 }

```

```

182 // executa a leitura das propriedades do livro pelo teclado
183 Livro leDoTeclado()
184 {
185     Livro livro;
186     cout << "Código: ";
187     cin >> livro.codigo;
188     cout << "Nome: ";
189     cin >> livro.nome;
190     return livro;
191 }
192
193 // responsável por exibir o menu e retornar a opção escolhida pelo usuário
194 int menu()
195 {
196     int opcao;
197     cout << "+++++ OPERAÇÕES +++++ " << endl;
198     cout << "1 - Inserir no início." << endl;
199     cout << "2 - Inserir em uma posição." << endl;
200     cout << "3 - Inserir no final." << endl;
201     cout << "4 - Remover no início." << endl;
202     cout << "5 - Remover em uma posição." << endl;
203     cout << "6 - Remover no final." << endl;
204     cout << "7 - Mostrar lista." << endl;
205     cout << "0 - Sair." << endl;
206     cout << "Digite: ";
207     cin >> opcao;
208     return opcao;
209 }

```