

```

1 //Alunos: Dericson Pablo e Lucas Henrique Fonseca
2 #include <iostream>
3 #define TAM 10
4
5 // tabela hashing como uma lista encadeada
6
7 using namespace std;
8
9 struct no{
10     int chave;
11     struct no *prox;
12 };
13
14 struct no *tabela[TAM];
15
16 void inicia(){
17     for(int i = 0; i < TAM; i++){
18         tabela[i] = NULL;
19     }
20 }
21
22 int espalha(int chave){
23     return chave % TAM;
24 }
25
26 void insere(int chave){
27     int pos;
28     struct no *atual, *anterior, *novo;
29     pos = espalha(chave);
30     atual = tabela[pos];
31
32     while(atual != NULL){
33         if(atual->chave == chave){
34             return; // pra nao inserir chave repetida
35         }
36         anterior = atual;
37         atual = atual->prox;
38     }
39
40     novo = new(struct no);
41     novo->chave = chave;
42     cout << "chave " << novo->chave << " inserida." << endl;
43     novo->prox = NULL;
44
45     if(tabela[pos] != NULL)
46         anterior->prox = novo;
47     else
48         tabela[pos] = novo;
49 }
50
51 void imprime(){
52     struct no *atual;
53     for(int i = 0; i < TAM; i++){
54         atual = tabela[i];
55         cout << "Posicao " << i << " da tabela = ";
56         if(atual == NULL){
57             cout << "vazia" << endl;
58         }else{
59             while(atual != NULL){
60                 cout << atual->chave << " ";
61                 atual = atual->prox;
62             }
63         }
64     }
65 }

```

```

67     }
68     cout << endl;
69 }
70 }
71 }
72
73 void busca(int chave){
74
75     int pos;
76     struct no *atual;
77     pos = espalha(chave);
78     atual = tabela[pos];
79     while (atual!=NULL){
80         if(atual -> chave == chave){
81             cout << "chave encontrada: " << chave << endl;
82             return;
83         }
84         atual = atual -> prox;
85     }
86     cout << "chave " << chave << " nao encontrada" << endl;
87
88 }
89
90 int main(){
91     inicia();
92     insere(50);
93     insere(51);
94     insere(52);
95     insere(53);
96     insere(54);
97     insere(55);
98     insere(56);
99     insere(57);
100    insere(58);
101    insere(68);
102    insere(59);
103    cout << "imprimindo.." << endl;
104    imprime();
105    cout << "buscando numeros..." << endl;
106    busca(23);
107    busca(68);
108    return 0;
109 }

```