

Nome: Messias Feres Curi Melo

Matrícula: 2022003764

Problema 1

main.c

```
C main.c > main()
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include "lista.h"
4
5  int main() {
6      int escolha, elemento;
7      Lista l;
8      inicializarLista(&l);
9
10     while (true){
11         printf("Menu\n");
12         printf("1 - Inserir\n");
13         printf("2 - Remoção\n");
14         printf("3 - Procurar\n");
15         printf("4 - Imprimir elementos\n");
16         printf("0 - Sair\n");
17         printf("Opção : ");
18         scanf("%d", &escolha);
19
20         if(escolha == 0){
21             break;
22         }else if (escolha == 1) {
23             printf("Elemento : ");
24             scanf("%d", &elemento);
25             inserirOrdenado(&l, elemento);
26             printf("Elemento inserido na lista\n");
27         }else if (escolha == 2) {
28             printf("Elemento : ");
29             scanf("%d", &elemento);
30             remover(&l, elemento);
31             printf("Elemento removido da lista\n");
32         }else if (escolha == 3) {
33             printf("Elemento : ");
34             scanf("%d", &elemento);
35             int indice = procura(&l, elemento);
36             if (indice != -1) {
37                 printf("Elemento encontrado no índice %d\n", indice);
38             }else {
39                 printf("Elemento não encontrado\n");
40             }
41         }else if (escolha == 4) {
42             printf("Elementos na lista: ");
43             for (int i = 0; i < l.tamanho; i++) {
44                 printf("%d ", l.elementos[i]);
45             }
46             printf("\n");
47         }else{
48             printf("Opção inválida\n");
49         }
50     }
51     return 0;
52 }
```

lista.c

```
C lista.c > inserirOrdenado(Lista *, int)
1  #include "lista.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void inicializarLista(Lista *l) {
6      l->tamanho = 0;
7  }
8
9  int procura(Lista *l, int elemento) {
10     for (int i = 0; i < l->tamanho; i++) {
11         if (l->elementos[i] == elemento) {
12             return i;
13         }
14     }
15     return -1;
16 }
17
18 void inserirOrdenado(Lista *l, int elemento) {
19     if (l->tamanho == TAM) {
20         return;
21     }
22     int i = l->tamanho - 1;
23     while (i >= 0 && l->elementos[i] > elemento) {
24         l->elementos[i + 1] = l->elementos[i];
25         i--;
26     }
27     l->elementos[i + 1] = elemento;
28     l->tamanho++;
29 }
30
31 int remover(Lista *l, int elemento) {
32     int indice = procura(l, elemento);
33     if (indice != -1) {
34         for (int i = indice; i < l->tamanho - 1; i++) {
35             l->elementos[i] = l->elementos[i + 1];
36         }
37         l->tamanho--;
38     }
39     return indice;
40 }
```

lista.h

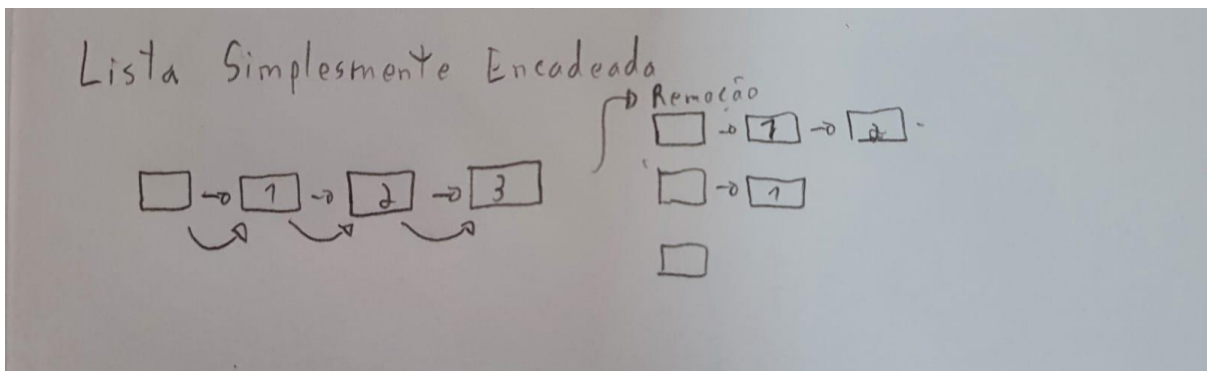
```
C lista.h > ...
1  # ifndef LISTA_H
2  # define LISTA_H
3
4  #define TAM 100
5
6  typedef struct Lista {
7      int elementos[TAM];
8      int tamanho;
9  }Lista;
10
11 void inicializarLista(Lista *l);
12 int procura(Lista *l, int elemento);
13 void inserirOrdenado(Lista *l, int elemento);
14 int remover(Lista *l, int elemento);
15
16 # endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n1$ gcc main.c lista.h lista.c -o n1
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n1$ ./n1
Menu
1 - Inserir
2 - Remoção
3 - Procurar
4 - Imprimir elementos
0 - Sair
Opção : 1
Elemento : 10
Elemento inserido na lista
Menu
1 - Inserir
2 - Remoção
3 - Procurar
4 - Imprimir elementos
0 - Sair
Opção : 1
Elemento : 3
Elemento inserido na lista
Menu
1 - Inserir
2 - Remoção
3 - Procurar
4 - Imprimir elementos
0 - Sair
Opção : 2
Elemento : 10
Elemento removido da lista
Menu
1 - Inserir
2 - Remoção
3 - Procurar
4 - Imprimir elementos
0 - Sair
Opção : 4
Elementos na lista: 3
Menu
1 - Inserir
2 - Remoção
3 - Procurar
4 - Imprimir elementos
0 - Sair
Opção : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n1$
```

Problema 2

Fotos



main.c

```
C main.c > main()
1  #include <stdio.h>
2  #include <stdbool.h>
3  #include "lista.h"
4
5  int main() {
6      int escolha, elemento;
7      Lista l;
8      inicializarLista(&l);
9
10     printf("Menu\n");
11     printf("1 - Tamanho\n");
12     printf("2 - Procura\n");
13     printf("3 - Inserir\n");
14     printf("4 - Remoção\n");
15     printf("5 - Imprimir elementos\n");
16     printf("0 - Sair\n");
17     while (true){
18         printf("Opção : ");
19         scanf("%d", &escolha);
20
21         if(escolha == 0){
22             break;
23         }else if (escolha == 1) {
24             int valor = tamanhoLista(&l);
25             printf("Tamanho da lista é : %d\n", valor);
26         }else if (escolha == 2) {
27             printf("Elemento : ");
28             scanf("%d", &elemento);
29             int indice = procura(&l, elemento);
30             if (indice != -1) {
31                 printf("Elemento encontrado no índice %d\n", indice);
32             }else {
33                 printf("Elemento não encontrado\n");
34             }
35         }else if (escolha == 3) {
36             printf("Elemento : ");
37             scanf("%d", &elemento);
38             inserirOrdenado(&l, elemento);
39             printf("Elemento inserido na lista\n");
40         }else if (escolha == 4) {
41             printf("Elemento : ");
42             scanf("%d", &elemento);
43             removerPrimeiraOcorrencia(&l, elemento);
44             printf("Elemento removido da lista\n");
45         }else if(escolha == 5) {
46             printf("Elementos na lista: ");
47             imprimirLista(&l);
48         }else{
49             printf("Opção inválida\n");
50         }
51     }
52     return 0;
53 }
```

lista.c

```
C lista.c > removePrimeiraOcorrencia(Lista *, int)
1  #include "lista.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void inicializarLista(Lista* l) {
6      l->cabeca = NULL;
7      l->tamanho = 0;
8  }
9
10 int tamanhoLista(Lista* l) {
11     return l->tamanho;
12 }
13
14 int procura(Lista* l, int elemento) {
15     No* atual = l->cabeca;
16     int indice = 0;
17     while (atual != NULL) {
18         if (atual->dado == elemento) {
19             return indice;
20         }
21         atual = atual->proximo;
22         indice++;
23     }
24     return -1;
25 }
26
27 void inserirOrdenado(Lista* l, int elemento) {
28     No* novoNo = (No*)malloc(sizeof(No));
29     if (novoNo == NULL) {
30         exit(1);
31     }
32     novoNo->dado = elemento;
33     novoNo->proximo = NULL;
34     if (l->cabeca == NULL || l->cabeca->dado >= elemento) {
35         novoNo->proximo = l->cabeca;
36         l->cabeca = novoNo;
37     } else {
38         No* anterior = NULL;
39         No* atual = l->cabeca;
40         while (atual != NULL && atual->dado < elemento) {
41             anterior = atual;
42             atual = atual->proximo;
43         }
44         anterior->proximo = novoNo;
45         novoNo->proximo = atual;
46     }
47     l->tamanho++;
48 }
49
50 void removerPrimeiraOcorrencia(Lista* l, int elemento) {
51     No* anterior = NULL;
52     No* atual = l->cabeca;
53
54     while (atual != NULL && atual->dado != elemento) {
55         anterior = atual;
56         atual = atual->proximo;
57     }
58
59     if (atual != NULL) {
60         if (anterior == NULL) {
61             l->cabeca = atual->proximo;
62         } else {
63             anterior->proximo = atual->proximo;
64         }
65         free(atual);
66         l->tamanho--;
67     }
68 }
69
70 void imprimirLista(Lista* l) {
71     No* atual = l->cabeca;
72     while (atual != NULL) {
73         printf("%d ", atual->dado);
74         atual = atual->proximo;
75     }
76     printf("\n");
77 }
```

lista.h

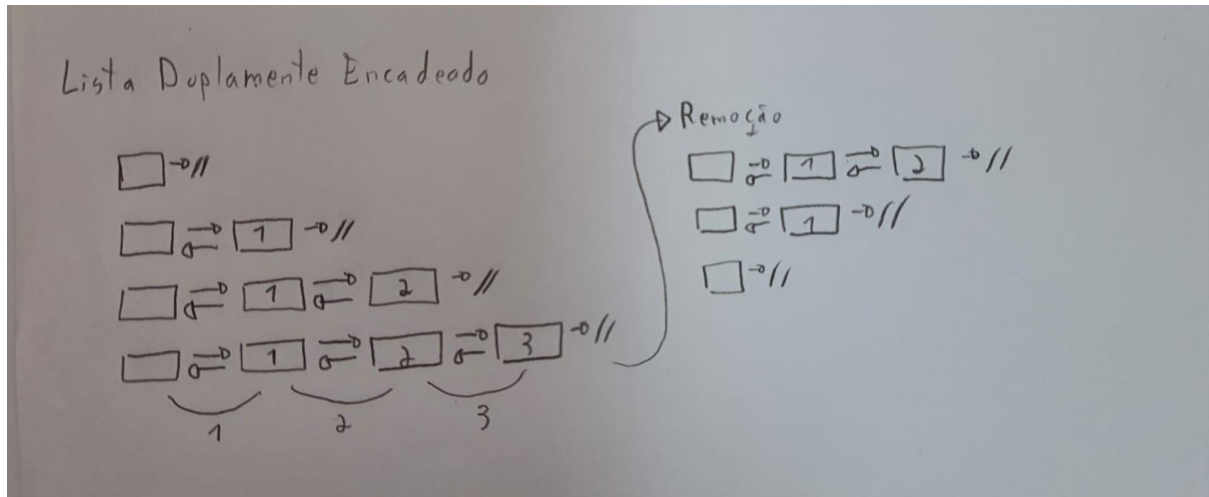
```
C lista.h > ...
1  # ifndef LISTA_H
2  # define LISTA_H
3
4  typedef struct No {
5      int dado;
6      struct No* proximo;
7  } No;
8
9  typedef struct Lista {
10     No* cabeca;
11     int tamanho;
12 } Lista;
13
14 void inicializarLista(Lista *l);
15 int tamanhoLista(Lista* l);
16 int procura(Lista* l, int elemento);
17 void inserirOrdenado(Lista* l, int elemento);
18 void removerPrimeiraOcorrencia(Lista* l, int elemento);
19 void imprimirLista(Lista* l);
20
21 # endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n2$ gcc main.c lista.h lista.c -o n2
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n2$ ./n2
Menu
1 - Tamanho
2 - Procura
3 - Inserir
4 - Remoção
5 - Imprimir elementos
0 - Sair
Opção : 3
Elemento : 12
Elemento inserido na lista
Opção : 3
Elemento : 23
Elemento inserido na lista
Opção : 3
Elemento : 4
Elemento inserido na lista
Opção : 2
Elemento : 5
Elemento não encontrado
Opção : 2
Elemento : 4
Elemento encontrado no índice 0
Opção : 4
Elemento : 23
Elemento removido da lista
Opção : 5
Elementos na lista: 4 12
Opção : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n2$
```

Problema 3

Fotos



main.c

```
C main.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "lista.h"
4
5 int main() {
6     int escolha, elemento;
7     Lista l;
8     inicializarLista(&l);
9
10    printf("Menu\n");
11    printf("1 - Tamanho\n");
12    printf("2 - Procura\n");
13    printf("3 - Inserir\n");
14    printf("4 - Remoção\n");
15    printf("5 - Imprimir elementos\n");
16    printf("0 - Sair\n");
17    while (true){
18        printf("Opção : ");
19        scanf("%d", &escolha);
20
21        if(escolha == 0){
22            break;
23        } else if (escolha == 1) {
24            int valor = tamanhoLista(&l);
25            printf("Tamanho da lista é : %d\n", valor);
26        } else if (escolha == 2) {
27            printf("Elemento : ");
28            scanf("%d", &elemento);
29            int indice = procura(&l, elemento);
30            if (indice != -1) {
31                printf("Elemento encontrado no índice %d\n", indice);
32            } else {
33                printf("Elemento não encontrado\n");
34            }
35        } else if (escolha == 3) {
36            printf("Elemento : ");
37            scanf("%d", &elemento);
38            inserirOrdenado(&l, elemento);
39            printf("Elemento inserido na lista\n");
40        } else if (escolha == 4) {
41            printf("Elemento : ");
42            scanf("%d", &elemento);
43            removerPrimeiraOcorrencia(&l, elemento);
44            printf("Elemento removido da lista\n");
45        } else if (escolha == 5) {
46            printf("Elementos na lista: ");
47            imprimirLista(&l);
48        } else {
49            printf("Opção inválida\n");
50        }
51    }
52    return 0;
53 }
```

lista.c

```
C lista.c > tamanhoLista(Lista *)
1  #include "lista.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void inicializarLista(Lista* l){
6      l->cabeca = NULL;
7      l->cauda = NULL;
8      l->tamanho = 0;
9  }
10
11 int tamanhoLista(Lista* l){
12     return l->tamanho;
13 }
14
15 int procura(Lista* l, int elemento){
16     No* atual = l->cabeca;
17     int indice = 0;
18     while(atual != NULL){
19         if(atual->dado == elemento){
20             return indice;
21         }
22         atual = atual->proximo;
23         indice++;
24     }
25     return -1;
26 }
27
28 void inserirOrdenado(Lista* l, int elemento){
29     No* novoNo = (No*)malloc(sizeof(No));
30     if(novoNo == NULL){
31         exit(1);
32     }
33     novoNo->dado = elemento;
34     novoNo->proximo = NULL;
35     novoNo->anterior = NULL;
36
37     if (l->cabeca == NULL || l->cabeca->dado >= elemento){
38         novoNo->proximo = l->cabeca;
39         if(l->cabeca != NULL){
40             l->cabeca->anterior = novoNo;
41         }else{
42             l->cauda = novoNo;
43         }
44         l->cabeca = novoNo;
45     }else{
46         No* atual = l->cabeca;
47         while (atual != NULL && atual->dado < elemento){
```



```

48     atual = atual->proximo;
49 }
50 if(atual != NULL){
51     novoNo->proximo = atual;
52     novoNo->anterior = atual->anterior;
53     atual->anterior = novoNo;
54     if(novoNo->anterior != NULL){
55         novoNo->anterior->proximo = novoNo;
56     }else{
57         l->cabeca = novoNo;
58     }
59 }else{
60     novoNo->anterior = l->cauda;
61     l->cauda->proximo = novoNo;
62     l->cauda = novoNo;
63 }
64 }
65 l->tamanho++;
66 }
67
68 void removerPrimeiraOcorrencia(Lista* l, int elemento){
69     No* atual = l->cabeca;
70     while(atual != NULL && atual->dado != elemento){
71         atual = atual->proximo;
72     }
73     if (atual != NULL){
74         if(atual->anterior != NULL){
75             atual->anterior->proximo = atual->proximo;
76         }else {
77             l->cabeca = atual->proximo;
78         }
79         if(atual->proximo != NULL){
80             atual->proximo->anterior = atual->anterior;
81         } else{
82             l->cauda = atual->anterior;
83         }
84         free(atual);
85         l->tamanho--;
86     }
87 }
88
89 void imprimirLista(Lista* l){
90     No* atual = l->cabeca;
91     while(atual != NULL){
92         printf("%d ", atual->dado);
93         atual = atual->proximo;
94     }
95     printf("\n");
96 }

```

lista.h

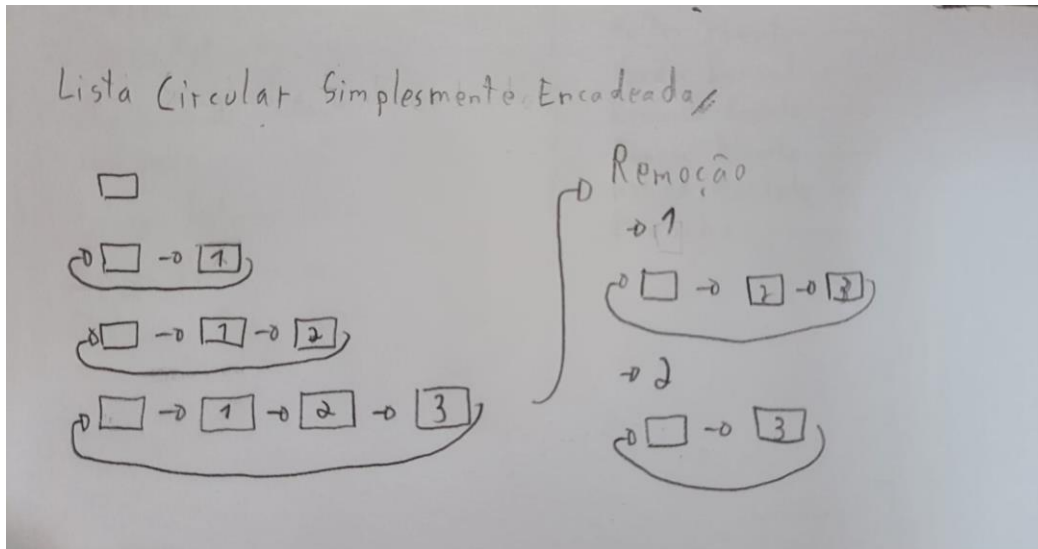
```
C lista.h > ...
1  # ifndef LISTA_H
2  # define LISTA_H
3
4  typedef struct No {
5      int dado;
6      struct No* proximo;
7      struct No* anterior;
8  } No;
9
10 typedef struct Lista {
11     No* cabeca;
12     No* cauda;
13     int tamanho;
14 } Lista;
15
16 void inicializarLista(Lista *l);
17 int tamanhoLista(Lista* l);
18 int procura(Lista* l, int elemento);
19 void inserirOrdenado(Lista* l, int elemento);
20 void removerPrimeiraOcorrencia(Lista* l, int elemento);
21 void imprimirLista(Lista* l);
22
23 # endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$ gcc main.c lista.h lista.c -o n3
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$ ./n3
Menu
1 - Tamanho
2 - Procura
3 - Inserir
4 - Remoção
5 - Imprimir elementos
0 - Sair
Opção : 3
Elemento : 12
Elemento inserido na lista
Opção : 3
Elemento : 1
Elemento inserido na lista
Opção : 3
Elemento : 253
Elemento inserido na lista
Opção : 2
Elemento : 1
Elemento encontrado no índice 0
Opção : 2
Elemento : 253
Elemento encontrado no índice 2
Opção : 4
Elemento : 1
Elemento removido da lista
Opção : 5
Elementos na lista: 12 253
Opção : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$
```

Problema 4

Fotos



main.c

```
C main.c > main()
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include "lista.h"
4
5 int main() {
6     int escolha, elemento;
7     Lista l;
8     inicializarLista(&l);
9
10    printf("Menu\n");
11    printf("1 - Tamanho\n");
12    printf("2 - Procura\n");
13    printf("3 - Inserir\n");
14    printf("4 - Remoção\n");
15    printf("5 - Imprimir elementos\n");
16    printf("0 - Sair\n");
17    while (true) {
18        printf("Opção : ");
19        scanf("%d", &escolha);
20
21        if (escolha == 0) {
22            break;
23        } else if (escolha == 1) {
24            int valor = tamanhoLista(&l);
25            printf("Tamanho da lista é : %d\n", valor);
26        } else if (escolha == 2) {
27            printf("Elemento : ");
28            scanf("%d", &elemento);
29            int indice = procura(&l, elemento);
30            if (indice != -1) {
31                printf("Elemento encontrado no índice %d\n", indice);
32            } else {
33                printf("Elemento não encontrado\n");
34            }
35        } else if (escolha == 3) {
36            printf("Elemento : ");
37            scanf("%d", &elemento);
38            inserirOrdenado(&l, elemento);
39            printf("Elemento inserido na lista\n");
40        } else if (escolha == 4) {
41            printf("Elemento : ");
42            scanf("%d", &elemento);
43            removerPrimeiraOcorrencia(&l, elemento);
44            printf("Elemento removido da lista\n");
45        } else if (escolha == 5) {
46            printf("Elementos na lista: ");
47            imprimirLista(&l);
48        } else {
49            printf("Opção inválida\n");
50        }
51    }
52    return 0;
53 }
```

lista.c

```
C lista.c > imprimirLista(Lista *)
1  #include "lista.h"
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  void inicializarLista(Lista* l){
6      l->cabeca = NULL;
7      l->tamanho = 0;
8  }
9
10 int tamanhoLista(Lista* l){
11     return l->tamanho;
12 }
13
14 int procura(Lista* l, int elemento){
15     if (l->cabeca != NULL) {
16         int indice = 0;
17         No* atual = l->cabeca;
18         do {
19             if (atual->dado == elemento) {
20                 return indice;
21             }
22             atual = atual->proximo;
23             indice++;
24         } while (atual != l->cabeca);
25     }
26     return -1;
27 }
28
29 void inserirOrdenado(Lista* l, int elemento) {
30     No* novoNo = (No*)malloc(sizeof(No));
31     if (novoNo == NULL) {
32         exit(1);
33     }
34     novoNo->dado = elemento;
35     novoNo->proximo = NULL;
36     if (l->cabeca == NULL || l->cabeca->dado >= elemento) {
37         novoNo->proximo = l->cabeca;
38         l->cabeca = novoNo;
39     } else {
40         No* anterior = NULL;
41         No* atual = l->cabeca;
42         while (atual != NULL && atual->dado < elemento) {
43             anterior = atual;
44             atual = atual->proximo;
45         }
46         anterior->proximo = novoNo;
47         novoNo->proximo = atual;
48     }
49     l->tamanho++;
50 }
51 void removerPrimeiraOcorrencia(Lista* l, int elemento) {
52     No* anterior = NULL;
53     No* atual = l->cabeca;
54
55     while (atual != NULL && atual->dado != elemento) {
56         anterior = atual;
57         atual = atual->proximo;
58     }
59
60     if (atual != NULL) {
61         if (anterior == NULL) {
62             l->cabeca = atual->proximo;
63         } else {
64             anterior->proximo = atual->proximo;
65         }
66         free(atual);
67         l->tamanho--;
68     }
69 }
70
71 void imprimirLista(Lista* l){
72     No* atual = l->cabeca;
73     while (atual != NULL) {
74         printf("%d ", atual->dado);
75         atual = atual->proximo;
76     }
77     printf("\n");
78 }
```

lista.h

```
C lista.h > ...
1  # ifndef LISTA_H
2  # define LISTA_H
3
4  typedef struct No {
5      int dado;
6      struct No* proximo;
7  } No;
8
9  typedef struct Lista {
10     No* cabeca;
11     int tamanho;
12 } Lista;
13
14 void inicializarLista(Lista *l);
15 int tamanhoLista(Lista* l);
16 int procura(Lista* l, int elemento);
17 void inserirOrdenado(Lista* l, int elemento);
18 void removerPrimeiraOcorrencia(Lista* l, int elemento);
19 void imprimirLista(Lista* l);
20
21 # endif
```

Console

```
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$ gcc main.c lista.h lista.c -o n4
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$ ./n4
Menu
1 - Tamanho
2 - Procura
3 - Inserir
4 - Remoção
5 - Imprimir elementos
0 - Sair
Opção : 3
Elemento : 1
Elemento inserido na lista
Opção : 3
Elemento : 23
Elemento inserido na lista
Opção : 3
Elemento : 16
Elemento inserido na lista
Opção : 4
Elemento : 16
Elemento removido da lista
Opção : 5
Elementos na lista: 1 23
Opção : 0
messiasfcm@MessiasFCM:/mnt/c/Users/Messi/OneDrive/Área de Trabalho/Lista4/n3$
```