

2018_1 - ALGORITMOS E ESTRUTURAS DE DADOS II - TN

PAINEL > **MINHAS TURMAS** > **2018 1 - ALGORITMOS E ESTRUTURAS DE DADOS II - TN**
> **VPL CHALLENGE 02: EXERCICIOS DE LISTAS/PILHAS - PROVA 2** > **P2.1 - BUSCA EM LISTA**

Descrição

[Visualizar envios](#)

P2.1 - Busca em lista

Data de entrega: terça, 1 Mai 2018, 23:55

Arquivos requeridos: lista.h, lista.c, main.c ([Baixar](#))

Tipo de trabalho: Trabalho individual

Crie um algoritmo que realize uma busca sequencial em uma lista.

A estrutura de sua lista deve salvar um ponteiro para a próxima célula e um inteiro. O algoritmo deve conter funções que:

1. Aloca uma lista vazia;
2. Insere um novo elemento na lista; Ou seja, tal função deveria inserir os N elementos da entrada;
3. Busque um elemento de forma sequencial em uma lista.

Você pode utilizar um .h similar ao abaixo.

```
#ifndef LISTA_H
#define LISTA_H

typedef struct node {
    int valor;
    struct node *next;
} node_t;

typedef struct {
    int num_elementos;
    node_t *inicio;
} lista_t;

lista_t *cria_lista();
void insere_valor(lista_t *lista, int valor);
int busca_valor(lista_t *lista, int valor);

#endif
```

Dica: Lembre-se que uma busca sequencial deve percorrer a lista, partindo do início até encontrar o elemento. Caso atinja o final da lista sem localizar o elemento, deve-se retornar o valor de -1.

A entrada é dada por N valores a serem inseridos em cada célula da lista e o elemento a ser buscado. Os valores devem ser inseridos na lista na ordem em que são dados.

A saída deve retornar à posição da lista a qual o elemento está localizado. Seu main pode ter a seguinte forma:

```
#include "lista.h"

int main(void) {
    // Aloca Lista
    // EXEMPLOS DE LEITURA DA ENTRADA
    while(scanf("%d", VARIABEL) != '\n')
    while(scanf("%d", VARIABEL) != EOF)

    // Insere elementos
    // Imprime saída
    return 0;
}
```

Arquivos requeridos

lista.h

```
1  #ifndef LISTA_H
2  #define LISTA_H
3
4  typedef struct node {
5      int valor;
6      struct node *next;
7  } node_t;
8
9  typedef struct {
10     int num_elementos;
11     node_t *inicio;
12 } lista_t;
13
14 lista_t *cria_lista();
15 void insere_valor(lista_t *lista, int valor);
16 int busca_valor(lista_t *lista, int valor);
17
18 #endif
```

lista.c

main.c

```
1  #include <stdio.h>
2
3  #include "lista.h"
4
5  int main(void) {
6
7      return 0;
8  }
9
```

[VPL](#)

[◀ Teste](#)

Seguir para...

[P2.2 - Remoção de duplicatas ▶](#)