

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** > **P2.3 - PALINDROMO**

Descrição

[Visualizar envios](#)

P2.3 - Palindromo

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 checa se uma lista é um palíndromo. Neste problema, faça uso de uma lista duplamente encadeada.

A estrutura de sua lista deve salvar um ponteiro para a próxima célula e um char. Logo, cada célula da lista representa uma letra de uma palavra.

O algoritmo deve conter funções que:

1. Crie uma lista vazia;
2. Insira um novo elemento em uma lista, neste caso, insira a letra na variável char da estrutura;
3. Verifica se a lista é um palíndromo.

Dica: Um palíndromo, é uma *string* que pode ser lida da esquerda para a direita da mesma forma que é lida da direita para a esquerda. Ou seja, as letras são espelhadas até certo ponto.

Exemplo de lista.h:

```
#ifndef LISTA_H
#define LISTA_H

typedef struct node {
    char valor;
    struct node *prox;
    struct node *prev;
} node_t;

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

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

#endif
```

A entrada é dada por N letras separadas por espaço que devem ser inseridas nas células da lista. A saída deve sinalizar se a lista é ou não um palíndromo. Caso a lista seja um palíndromo retorne:

1

Caso a lista não seja um palíndromo retorne:

0

Exemplo de main:

```
#include "lista.h"

int main(void) {
    // Aloca Lista
    // EXEMPLOS DE LEITURA DA ENTRADA
    while(scanf(" %c", VARIAVEL) != '\n`)
    // OU
    while (scanf(" %c", VARIAVEL) != 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      char valor;
6      struct node *prox;
7      struct node *prev;
8  } node_t;
9
10 typedef struct {
11     int num_elementos;
12     node_t *inicio;
13     node_t *fim;
14 } lista_t;
15
16 lista_t *cria_lista();
17 void insere_valor(lista_t *lista, char valor);
18 int palindromo(lista_t *lista);
19
20 #endif
```

lista.c

main.c

```
1  #include "lista.h"
2
3  int main(void) {
4      // Lê entrada
5      // Para ler a entrada insira um espaço no scanf(" %c", variavel)
6      // Aloca lista
7      // Insere elementos
8      // Checa palindromo e retorna valor
9      return 0;
10 }
```

[VPL](#)

◀ [P2.2 - Remoção de duplicatas](#)

Seguir para...

[P2.3.1 Palindromo V2](#) ▶