

Segunda Prova de Algoritmos e Estruturas de Dados I

08/05/2026

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: O que será avaliado?
R: Especialmente nesta prova, vetores e modularidade: uso correto de funções e procedimentos, passagem de parâmetros (valor vs. referência) e uso de variáveis locais. Também serão pontuados: lógica, clareza, sintaxe, nomes de variáveis significativos, indentação e uso racional de comentários.

Boa prova!!

1) [40 pontos] Escreva um programa completo em Free Pascal que, dado um vetor de inteiros, o rotacione uma posição à esquerda. Por exemplo, rotacionar [3 1 4 2 7] uma posição à esquerda resulta em [1 4 2 7 3]. Em seguida, determine se o vetor resultante é um palíndromo, ou seja, se ele lido da esquerda para a direita é igual ao lido da direita para a esquerda.

O programa deve conter:

- Um **procedimento** para rotacionamento que receba o vetor e seu tamanho útil e modifique o vetor original rotacionando-o uma posição à esquerda;
- Uma **função** que receba o vetor resultante do passo anterior e seu tamanho útil e retorne **true** se o vetor for palíndromo, ou **false** caso contrário;
- O **programa principal** que leia o tamanho útil do vetor e em seguida chame um procedimento de leitura com a mesma estrutura de **preenche_vetor** (que será escrito na Questão 2), invoque os subprogramas descritos acima, imprima o vetor rotacionado e, em seguida, a mensagem **Eh palindromo** ou **Nao eh palindromo**.

Exemplos de execução:

Entrada	Vetor rotacionado	Saída
v = [3 1 4 2 7]	[1 4 2 7 3]	Nao eh palindromo
v = [1 1 2 3 2]	[1 2 3 2 1]	Eh palindromo
v = [5 3 2 4 1]	[3 2 4 1 5]	Nao eh palindromo

2) [60 pontos] Escreva um programa completo em Free Pascal que identifique se um vetor possui um **ponto de equilíbrio**. Um índice i é considerado um ponto de equilíbrio se a soma de todos os elementos estritamente à sua esquerda for igual à soma de todos os elementos estritamente à sua direita.

Para esta definição, considere que:

- O elemento contido no próprio índice i não integra nenhuma das duas somas;
- Caso não existam elementos à esquerda (quando $i = 1$) ou à direita (quando $i = N$), a soma correspondente é considerada zero;

- Caso existam múltiplos pontos de equilíbrio, o programa deve retornar o de menor índice.

O programa deve conter:

- Um **procedimento preenche_vetor** que preenche um vetor de, no máximo, 100 números inteiros do teclado;
- Uma **função** que receba o vetor e seu tamanho útil. O retorno deve ser o índice do ponto de equilíbrio encontrado, ou -1 caso ele não exista;
- O **programa principal** que leia o tamanho útil do vetor e em seguida invoque os subprogramas acima e imprima o índice encontrado ou a mensagem Nao ha ponto de equilibrio.

Exemplos de execução:

Entrada	Saída
v = [1 7 3 6 5 6]	Ponto de equilibrio no indice 4
v = [10 5 2]	Nao ha ponto de equilibrio