

Lista 1 Monitoria ED - Algoritmos e Ponteiros

Questão 1 - Algoritmos:

Crie um programa que leia os três coeficientes de uma equação de segundo grau, a , b e c , e imprima o valor das raízes. Calcule as raízes se o valor do discriminante (Δ) for maior ou igual a zero. Se for menor do que zero, apenas imprima uma mensagem adequada e finalize o programa.

Questão 2 - Algoritmos:

Partindo-se de um único casal de coelhos filhotes recém-nascidos, supondo-se que um casal de coelhos se torne fértil após dois meses de vida e a partir de então, produz um novo casal a cada mês e assumindo-se que os coelhos nunca morrem, a quantidade de casal de coelhos após n meses é dado pelo n -ésimo termo da seguinte sequência:

$$F_n = F_{n-2} + F_{n-1}, n \geq 2 \quad F_0 = 0 \\ F_1 = 1.$$

Esta sequência é chamada de sequência de Fibonacci. Faça um programa que mostre a quantidade de casais de coelhos após n meses, onde n é dado pelo usuário.

(Para quem conseguir implementar de maneira rápida: Faça uma versão de código que implemente uma função para calcular Fibonacci de forma **recursiva**)

Questão 3 - Ponteiros:

Crie um programa que leia um vetor de 10 números inteiros e retorne o maior e o menor número do vetor. O resultado deve ser exibido em *main*, utilizando variáveis criadas em *main*. Todavia, o cálculo deve ser feito em uma função, que deve ter o seguinte escopo:

*void calcula_min_max(int *vetor, int *maior, int *menor)*

onde “maior” e “menor” devem receber os endereços das variáveis em *main* que guardarão os resultados para serem exibidos posteriormente.

Questão 4 - Ponteiros:

Escreva uma função que receba um vetor de números reais e tenha como valor de retorno um novo vetor, alocado dinamicamente, com os elementos do vetor original em ordem reversa. A função deve ter como valor de retorno o ponteiro do vetor alocado, seguindo o protótipo:

float reverter_vetor (int n, float *v)*

Faça uma função *main* para testar sua função. Na função *main*, não esqueça de liberar a memória alocada pela função auxiliar.

Questão 5 - Ponteiros:

Escreva uma função que receba como parâmetros duas strings e um caractere separador. A função deve criar a string que representa a concatenação das duas strings de entrada, usando o caractere como separador. Por exemplo, se forem passadas as strings “ex” e “aluno”, e o caractere hífen ‘-’, deve-se ter como valor de retorno a string “ex-aluno”. O protótipo da função deve ser:

char concatena (char *s1, char *s2, char sep);*