Claro! Abaixo estão as soluções em linguagem C para as 11 questões propostas, acompanhadas de uma breve explicação de cada lógica para facilitar sua revisão para a prova:

**QUESTÃO 01: Contar ocorrências de um número x em um vetor A**

**Lógica:** Percorrer o vetor e contar quantas vezes o número x aparece.

int contar\_ocorrencias(int A[], int n, int x) {

int cont = 0;

for (int i = 0; i < n; i++) {

if (A[i] == x)

cont++;

}

return cont;

}

**QUESTÃO 02: Trocar pares de elementos consecutivos em um vetor de reais**

**Lógica:** Percorrer o vetor em passos de 2 e trocar cada par de elementos consecutivos.

void trocar\_pares(float vetor[], int n) {

for (int i = 0; i < n - 1; i += 2) {

float temp = vetor[i];

vetor[i] = vetor[i + 1];

vetor[i + 1] = temp;

}

}

**QUESTÃO 03: Verificar se um vetor está ordenado de forma crescente**

**Lógica:** Comparar cada elemento com o próximo; se algum elemento for maior que o seguinte, o vetor não está ordenado.

int esta\_crescente(float vetor[], int n) {

for (int i = 0; i < n - 1; i++) {

if (vetor[i] > vetor[i + 1])

return 0;

}

return 1;

}

**QUESTÃO 04: Gerar os 50 primeiros termos da sequência S = 1, 2, 4, 7, 11, 16, ...**

**Lógica:** Iniciar com o primeiro termo e, a cada passo, adicionar um incremento que aumenta em 1 a cada iteração.

void gerar\_sequencia(int S[], int n) {

S[0] = 1;

int incremento = 1;

for (int i = 1; i < n; i++) {

S[i] = S[i - 1] + incremento;

incremento++;

}

}

**QUESTÃO 05: Separar 300 valores inteiros em vetores de pares e ímpares**

**Lógica:** Iterar sobre os 300 valores e, utilizando o operador módulo, separar os números em pares e ímpares.

void separar\_pares\_impares(int valores[], int n, int pares[], int \*n\_pares, int impares[], int \*n\_impares) {

\*n\_pares = \*n\_impares = 0;

for (int i = 0; i < n; i++) {

if (valores[i] % 2 == 0)

pares[(\*n\_pares)++] = valores[i];

else

impares[(\*n\_impares)++] = valores[i];

}

}

**QUESTÃO 06: Substituir todas as ocorrências do número A pelo número B em um vetor de reais**

**Lógica:** Percorrer o vetor e substituir cada ocorrência de A por B.

void substituir\_elementos(float vetor[], int n, float A, float B) {

for (int i = 0; i < n; i++) {

if (vetor[i] == A)

vetor[i] = B;

}

}

**QUESTÃO 07: Remover todas as ocorrências de um elemento de um vetor de float e retornar o número de remoções**

**Lógica:** Filtrar o vetor para remover todas as ocorrências do elemento e calcular a diferença de tamanho para obter o número de remoções.

int remover\_elemento(float vetor[], int \*n, float elemento) {

int i, j = 0, removidos = 0;

for (i = 0; i < \*n; i++) {

if (vetor[i] != elemento)

vetor[j++] = vetor[i];

else

removidos++;

}

\*n = j;

return removidos;

}

**QUESTÃO 08: Encontrar a posição de x em um vetor ordenado ou a posição onde deveria estar**

**Lógica:** Utilizar busca binária para encontrar a posição de x ou determinar onde ele deveria ser inserido para manter a ordem.

int buscar\_posicao(float vetor[], int n, float x) {

int inicio = 0, fim = n - 1, meio;

while (inicio <= fim) {

meio = (inicio + fim) / 2;

if (vetor[meio] == x)

return meio;

else if (vetor[meio] < x)

inicio = meio + 1;

else

fim = meio - 1;

}

return inicio; // posição onde x deveria estar

}

**QUESTÃO 09: Inserir um elemento em um vetor ordenado crescentemente**

**Lógica:** Utilizar a função da questão 08 para encontrar a posição correta e inserir o elemento.

void inserir\_ordenado(float vetor[], int \*n, float x) {

int pos = buscar\_posicao(vetor, \*n, x);

for (int i = \*n; i > pos; i--) {

vetor[i] = vetor[i - 1];

}

vetor[pos] = x;

(\*n)++;

}

**QUESTÃO 10: União de dois vetores A e B sem repetição**

**Lógica:** Adicionar todos os elementos de A ao vetor C e, em seguida, adicionar os elementos de B que não estão em A.

int uniao\_vetores(int A[], int nA, int B[], int nB, int C[]) {

int nC = 0;

for (int i = 0; i < nA; i++) {

C[nC++] = A[i];

}

for (int i = 0; i < nB; i++) {

int j;

for (j = 0; j < nA; j++) {

if (B[i] == A[j])

break;

}

if (j == nA)

C[nC++] = B[i];

}

return nC;

}

**QUESTÃO 11: Interseção de dois vetores A e B**

**Lógica:** Adicionar ao vetor C os elementos que estão presentes tanto em A quanto em B.

int intersecao\_vetores(int A[], int nA, int B[], int nB, int C[]) {

int nC = 0;

for (int i = 0; i < nA; i++) {

for (int j = 0; j < nB; j++) {

if (A[i] == B[j]) {

C[nC++] = A[i];

break;

}

}

}

return nC;

}

Espero que essas soluções e explicações auxiliem em sua preparação para a prova. Se precisar de mais alguma coisa, estou à disposição!