(fluxograma e pseudocódigo).

**QUESTÃO 2:**

**PSEUDOCODIGO**

Função bubble\_sort(vetor, n)

Para k de 0 até n - 1

Para j de 0 até n - k - 1

Se vetor[j] > vetor[j + 1]

Trocar(vetor[j], vetor[j + 1])

Função calculaDesvioPadrao(array, media, n)

desvio = 0

Para cada elemento no array

desvio += (elemento - media) ^ 2

Retorna raiz quadrada de (desvio / n)

Função calcularMedianaOrdenado(array, n)

Se n for par

Retorna (array[n / 2 - 1] + array[n / 2]) / 2.0

Senão

Retorna array[n / 2]

Função apresentarValoresRepetidos(array, n)

Crie um array frequencia[N] inicializado com zeros

Para cada elemento no array

frequencia[elemento]++

Para cada índice i em N

Se frequencia[i] > 1

Exiba "Valor i repetido frequencia[i] vezes"

Crie um novoArray

contador = 0

Para cada índice i em N

Se frequencia[i] > 0

novoArray[contador] = i

contador++

Exiba "Novo Array sem repetições (ordenado):"

Para cada elemento em novoArray

Exiba elemento

media = 0

Para cada elemento em novoArray

media += elemento

media /= contador

bubble\_sort(novoArray, contador)

mediana = calcularMedianaOrdenado(novoArray, contador)

Exiba "Média:", media

Exiba "Mediana:", mediana

Exiba "Desvio Padrão:", calculaDesvioPadrao(novoArray, media, contador)

**Função principal()**

Declare array[N], maior[P], menor[P], Pos\_MaiorV[P], Pos\_MenorV[P]

Inicialize variáveis media, mediana, desvio\_padrao

Preencha array com valores aleatórios entre 0 e N

Para cada elemento no array

Encontre os três maiores e menores valores e suas posições

Exiba "Os 3 Maiores Numeros:"

Para cada índice j em P

Exiba Pos\_MaiorV[j], maior[j]

Exiba "Os 3 Menores Numeros:"

Para cada índice j em P

Exiba Pos\_MenorV[j], menor[j]

media = 0

Para cada elemento no array

media += elemento

media /= N

bubble\_sort(array, N)

Exiba "Vetor Ordenado:"

Para cada elemento no array

Exiba elemento

mediana = calcularMedianaOrdenado(array, N)

Exiba "Média:", media

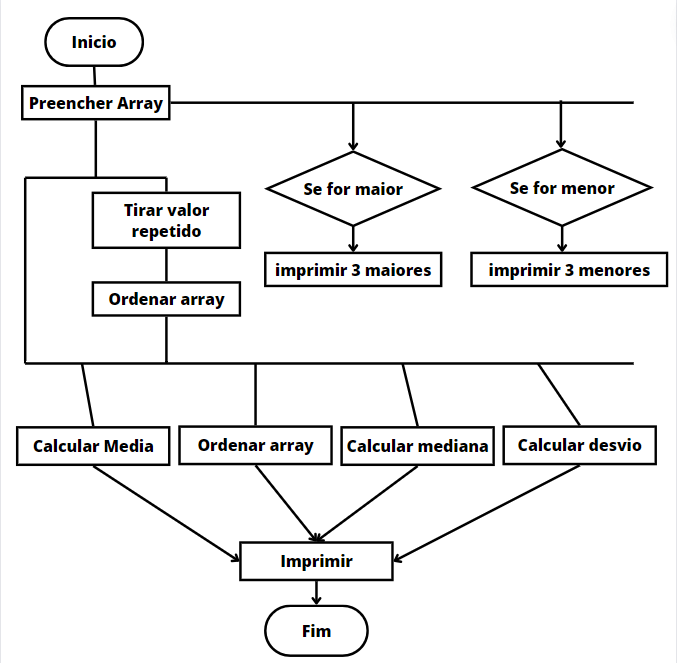
Exiba "Mediana:", mediana

desvio\_padrao = calculaDesvioPadrao(array, media, N)

Exiba "Desvio Padrão:", desvio\_padrao

Chame apresentarValoresRepetidos(array, N)

FLUXOGRAMA:



**QUESTÃO 3:**

**PSEUDOCODIGO**

Função bubble\_sort(vetor, n)

Para k de 0 até n - 1

Para j de 0 até n - k - 1

Se vetor[j] > vetor[j + 1]

Trocar(vetor[j], vetor[j + 1])

Função calculaDesvioPadrao(array, media, n)

desvio = 0

Para cada elemento no array

desvio += (elemento - media) ^ 2

Retorna raiz quadrada de (desvio / n)

Função calcularMedianaOrdenado(array, n)

Se n for par

Retorna (array[n / 2 - 1] + array[n / 2]) / 2.0

Senão

Retorna array[n / 2]

Função apresentarValoresRepetidos(array, n)

Crie um array novoArray

index = 0

Para cada elemento no array

Para cada elemento no novoArray até o index atual

Se elemento é igual a novoArray[i]

Exiba "Número repetido: i: elemento"

Interrompa o loop interno

Se não encontrou duplicatas

Adicione o elemento ao novoArray e incremente o index

Exiba "Novo array sem repetições (ordenado):"

Para cada elemento no novoArray até o index atual

Exiba "i: elemento"

media = 0

Para cada elemento no novoArray até o index atual

media += elemento

media /= index

Exiba "Média:", media

bubble\_sort(novoArray, index)

mediana = calcularMedianaOrdenado(novoArray, index)

Exiba "Mediana:", mediana

desvio\_padrao = calculaDesvioPadrao(novoArray, media, index)

Exiba "Desvio Padrão:", desvio\_padrao

**Função principal()**

Declare array[N], maior[P], menor[P], Pos\_MaiorV[P], Pos\_MenorV[P] em FLOAT

Inicialize variáveis media, mediana, desvio\_padrao

Preencha array com valores aleatórios entre 0 e MIN

Exiba os valores aleatórios

Para cada elemento no array

Encontre os três maiores e menores valores e suas posições

Exiba "Os 3 Maiores Numeros:"

Para cada índice j em P

Exiba "Pos\_MaiorV[j]: maior[j]"

Exiba "Os 3 Menores Numeros:"

Para cada índice j em P

Exiba "Pos\_MenorV[j]: menor[j]"

media = 0

Para cada elemento no array

media += elemento

media /= N

bubble\_sort(array, N)

Exiba "Vetor Ordenado:"

Para cada elemento no array

Exiba "i: elemento"

mediana = calcularMedianaOrdenado(array, N)

Exiba "Média:", media

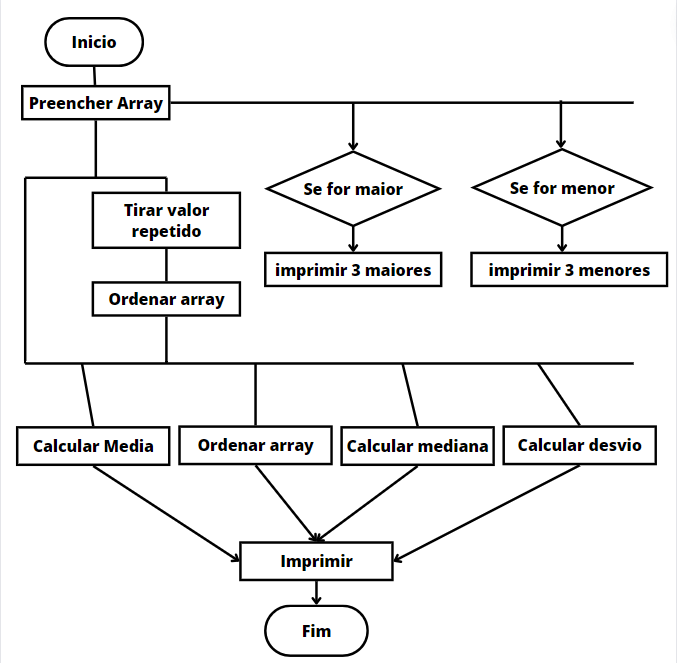
Exiba "Mediana:", mediana

desvio\_padrao = calculaDesvioPadrao(array, media, N)

Exiba "Desvio Padrão:", desvio\_padrao

Chame apresentarValoresRepetidos(array, N)

FLUXOGRAMA:



**QUESTÃO 4:**

**PSEUDOCODIGO**

Função bubble\_sort(vetor, n)

Para k de 0 até n - 1

Para j de 0 até n - k - 1

Se vetor[j] > vetor[j + 1]

Trocar(vetor[j], vetor[j + 1])

Função calculaDesvioPadrao(array, media, n)

desvio = 0

Para cada linha i no array

Para cada coluna j no array

desvio += (array[i][j] - media) ^ 2

Retorna raiz quadrada de (desvio / (n \* n))

Função calcularMedianaOrdenado(array, n)

Crie um vetor unidimensional auxiliar e copie todos os elementos do array para ele

Ordene o vetor auxiliar usando bubble\_sort

Se n for par

Retorna (auxiliar[n / 2 - 1] + auxiliar[n / 2]) / 2.0

Senão

Retorna auxiliar[n / 2]

Função apresentarValoresRepetidos(array, n)

Crie um novoArray para armazenar valores únicos

index = 0

Para cada linha i no array

Para cada coluna j no array

Verifique se array[i][j] já está em novoArray

Se não estiver, adicione array[i][j] a novoArray e incremente o index

Se estiver, exiba "Número repetido: [i][j]: array[i][j]"

Exiba "Novo Array:"

Para cada elemento no novoArray

Exiba elemento

Calcule média, mediana e desvio padrão usando funções auxiliares

Exiba média, mediana e desvio padrão

**Função principal()**

Declare array[N][N], maior[P], menor[P], Pos\_MaiorV[P][2], Pos\_MenorV[P][2]

Inicialize variáveis media, mediana, desvio\_padrao

Preencha array com valores aleatórios entre 0 e MIN

Exiba a matriz normal

Para cada elemento no array, encontre os três maiores e menores valores e suas posições

Exiba "Os 3 Maiores Numeros:"

Para cada índice j em P

Exiba "Pos\_MaiorV[j][0][1]: maior[j]"

Exiba "Os 3 Menores Numeros:"

Para cada índice j em P

Exiba "Pos\_MenorV[j][0][1]: menor[j]"

Calcule média, mediana e desvio padrão da matriz

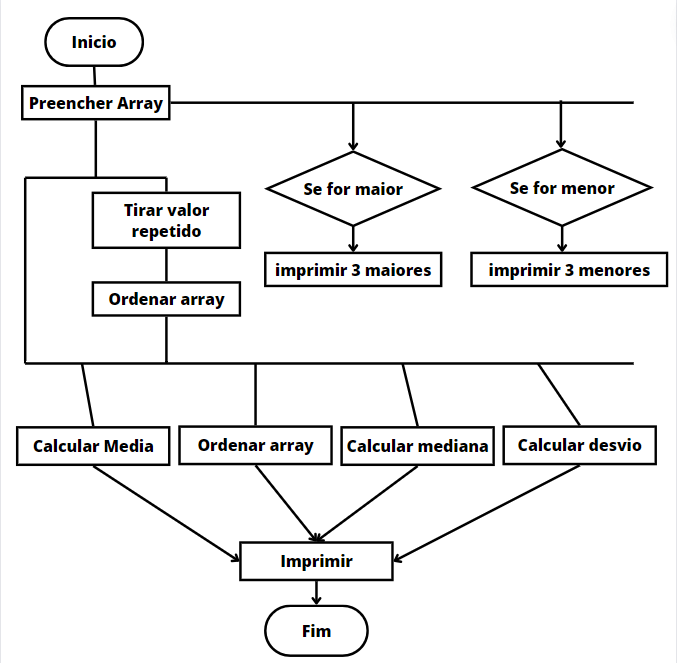
Exiba "Média:", media

Exiba "Mediana:", mediana

Exiba "Desvio Padrão:", desvio\_padrao

Chame apresentarValoresRepetidos(array, N)

FLUXOGRAMA:



**QUESTÃO 5:**

**PSEUDOCODIGO**

Função calcularSomatorioLinhasColunas(array, somatorioLinha, somatorioColuna, n)

Para cada linha i em array

Zerar somatorioLinha[i] e somatorioColuna[i]

Para cada coluna j em array

somatorioLinha[i] += array[i][j]

somatorioColuna[j] += array[i][j]

Função apresentarValoresRepetidos(array, n)

Crie um novoArray para armazenar valores únicos

index = 0

Para cada linha i em array

Para cada coluna j em array

Verifique se array[i][j] já está em novoArray

Se não estiver, adicione array[i][j] a novoArray e incremente o index

Se estiver, exiba "Número repetido: [%d][%d]: %.2f"

Exiba "Novo Array:"

Para cada elemento no novoArray

Exiba "[%d]: %.2f"

Calcule média, mediana e desvio padrão usando funções auxiliares

Exiba "Média: %.2f"

Exiba "Mediana: %.2f"

Exiba "Desvio Padrão: %.2f"

**Função principal()**

Declare array[N][N], maior[P], menor[P], Pos\_MaiorV[P][2], Pos\_MenorV[P][2]

Declare variáveis media, mediana, desvio\_padrao

Declare vetores somatorioLinha[N] e somatorioColuna[N]

Preencha array com valores aleatórios entre 0 e MIN

Exiba a matriz normal

Calcule e exiba os 3 maiores e menores valores e suas posições

Calcule média, mediana e desvio padrão da matriz

Exiba "Média:", media

Exiba "Mediana:", mediana

Exiba "Desvio Padrão:", desvio\_padrao

Calcule os somatórios das linhas e colunas

Exiba "Somatorio das Linhas:"

Para cada elemento em somatorioLinha

Exiba "[%d]: %.2f"

Exiba "Somatorio das Colunas:"

Para cada elemento em somatorioColuna

Exiba "[%d]: %.2f"

Calcule média, mediana e desvio padrão dos somatórios das linhas e colunas

Exiba "Media das Somatorias das Linhas: %.2f"

Exiba "Mediana das Somatorias das Linhas: %.2f"

Exiba "Desvio Padrão das Somatorias das Linhas: %.2f"

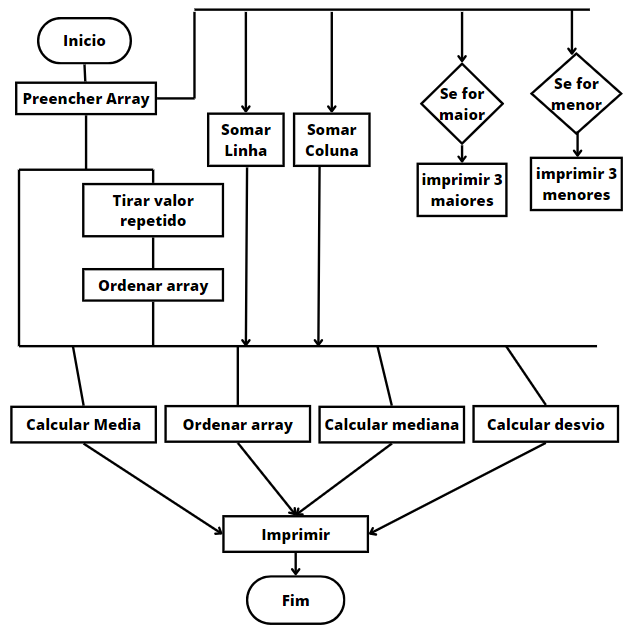
Exiba "Media das Somatorias das Colunas: %.2f"

Exiba "Mediana das Somatorias das Colunas: %.2f"

Exiba "Desvio Padrão das Somatorias das Colunas: %.2f"

Chame apresentarValoresRepetidos(array, N)

FLUXOGRAMA:



**QUESTÃO 6:**

**PSEUDOCODIGO**

**INÍCIO**

// Definir constantes

const N = 10

const P = 20

const O = 5

// Declarar array tridimensional

inteiro array[N][P][O]

// Inicializar contador

inteiro contador = 0

// Preencher o array tridimensional

PARA cada i DE 0 ATÉ N FAÇA

PARA cada j DE 0 ATÉ P FAÇA

PARA cada k DE 0 ATÉ O FAÇA

array[i][j][k] = contador

contador = contador + 1

FIM PARA

FIM PARA

FIM PARA

// Exibir o array tridimensional

PARA cada i DE 0 ATÉ N FAÇA

PARA cada j DE 0 ATÉ P FAÇA

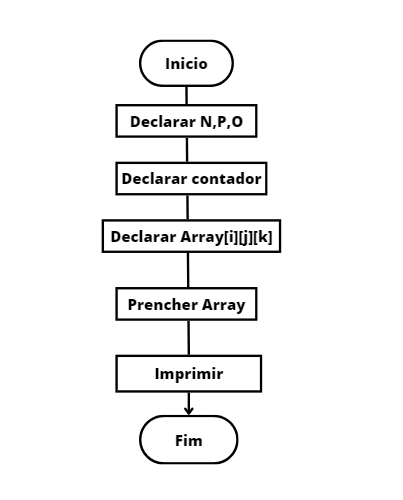
PARA cada k DE 0 ATÉ O FAÇA

ESCREVER "[" + i + "][" + j + "][" + k + "]: " + array[i][j][k]

FIM PARA

FIM PARA

FIM PARA

FIM

FLUXOGRAMA:

**QUESTÃO 7:**

**PSEUDOCODIGO**

Função: printarTabuleiro

Para cada linha no tabuleiro

Para cada coluna na linha

Imprimir elemento na posição (linha, coluna)

Fim Para

Imprimir quebra de linha

Fim Para

Fim Função

Função: inicializarTabuleiro

Para cada linha no tabuleiro

Para cada coluna na linha

Atribuir '\_' ao elemento na posição (linha, coluna)

Fim Para

Fim Para

Fim Função

Função: adicionarPista

Para cada letra na resposta da pista

Se direção da pista é 'H'

Atribuir a letra ao elemento na posição (linha, coluna + índice)

Senão Se direção da pista é 'V'

Atribuir a letra ao elemento na posição (linha + índice, coluna)

Fim Se

Fim Para

Fim Função

Função: completarPalavra

Enquanto a palavra inserida pelo usuário não é igual à resposta da pista

Pedir ao usuário para inserir a palavra

Fim Enquanto

Para cada letra na resposta da pista

Se direção da pista é 'H'

Atribuir a letra inserida pelo usuário ao elemento na posição (linha, coluna + índice)

Senão Se direção da pista é 'V'

Atribuir a letra inserida pelo usuário ao elemento na posição (linha + índice, coluna)

Fim Se

Fim Para

Fim Função

**INICIO**

Declaração: tabuleiro[SIZE][SIZE], tabuleiroInicial[SIZE][SIZE]

Atribuir um tabuleiro inicial predefinido a tabuleiroInicial

Escrever "Palavra Cruzada:"

Chamada: printarTabuleiro(tabuleiroInicial)

Chamar: inicializarTabuleiro(tabuleiro)

Para cada pista nas pistas de palavras cruzadas

Chamar: adicionarPista(tabuleiro, pista)

Fim Para

Para cada pista nas pistas de palavras cruzadas

Chamar: completarPalavra(tabuleiro, pista)

Fim Para

Escrever "Tabuleiro Final:"

Chamar: printarTabuleiro(tabuleiro)

Fim Função

FLUXOGRAMA:

