



Nexaas

TECNOLOGIA DA INFORMAÇÃO
Lógica de Programação

Paulo S R Rios
paulinhorios@gmail.com



Vetores: Podem ser vistos como uma variável que possui diversas posições, porém todos os dados armazenados neste tipo de variável são do mesmo tipo.

Inteiro vetor_numeros[5]

// declaração de uma variável do tipo vetor inteiro com 6 posições ([0],[1],[2],[3],[4],[5])

real vetor_valor[3] = {1, 2.4, 3, 4.7}

//declaração de variável do tipo vetor real com 4 posições inicializada ([0],[1],[2],[3])



Vetores: Os elementos individuais são acessados por sua posição no vetor. Como um vetor tem mais de uma posição, deve-se indicar qual posição do vetor que deseja acessar. Nesse caso usamos um índice.

A posição inicial do vetor inicia no índice 0 (zero)

```
real vetor_valor[3] = {1, 2.4, 3, 4.7}
```

```
escreva(vetor_valor[2])
```

```
// exibido na tela o valor real 3
```

* o escreva vai exibir o conteúdo da posição 2 do índice que é igual ao terceiro elemento do vetor (3)



Exercício 1: Faça um algoritmo que armazene 10 nomes referente a lista de presença em uma fila de atendimento. Crie um vetor de 10 posições onde cada índice armazene um nome na posição de chegada. Ao final a lista deve ser exibida com todos os nomes armazenados.

*Utilize uma estrutura de repetição para armazenar as posições e para exibir a lista.



Exercício 2: Faça um algoritmo que armazene as leituras pluviométricas de uma estação meteorológica de uma determinada localidade no mês de dezembro de 2022 (total de 31 dias).

Cada dia receberá uma leitura total (valores do tipo real) de mm de chuva. Ao final da leitura o programa deve exibir o somatório de todas as leituras e exibir o total de chuva no mês de dezembro.

*100mm de chuva significa que em uma área de 1m^2 , a lâmina de água formada pela chuva apresenta uma altura de 100 milímetros.



Matriz: A matriz é definida como sendo um vetor com mais de uma dimensão (geralmente duas). Enquanto o vetor armazena as informações de forma linear, a matriz armazena de forma tabular (Linhas e Colunas)

inteiro matriz_numeros[2][2]

A matriz acima representa uma matriz de 3 linhas e 3 colunas (lembrando que o índice inicia na posição 0)



Matriz: As posições são acessadas de acordo com o código das coordenadas da matriz.

```
matriz_numeros[0][0] = 1  
matriz_numeros[0][1] = 2  
matriz_numeros[0][2] = 3  
matriz_numeros[1][0] = 4  
matriz_numeros[1][1] = 5  
matriz_numeros[1][2] = 6  
matriz_numeros[2][0] = 7  
matriz_numeros[2][1] = 8  
matriz_numeros[2][2] = 9
```

1	2	3
4	5	6
7	8	9



Exercício 3: Faça um algoritmo que simule o jogo batalha naval. O mapa da batalha tem 25 posições (5x5), cada posição pode armazenar textos referente aos alvos atingidos exemplo: água, cruzador, porta-aviões, fragata, couraçado,...).

Inicialmente um dos jogadores deve inserir as 25 posições com os nomes dos alvo.

Após a leitura das posições outro jogador tem 3 chances de acertar um alvo informando a linha e a coluna e o programa deve exibir o que foi atingido a cada tentativa.



Exercício 4: Atualize o exercício anterior com um menu inicial para o jogo com as opções como abaixo:

- 1 – Inserir Frota no mapa da Batalha
- 2 – Exibir Frota no mapa da Batalha
- 3 – Iniciar Combate (3 disparos)
- 4 – Sair do Programa



Exercício 5: Faça um algoritmo que simule a venda de ingressos para o cinema. As poltronas são numeradas em filas (de 1 a 10) e cadeiras (de 1 a 20).

O algoritmo deve ter um menu como abaixo:

- 1 – Vender Ingresso
- 2 – Cancelar venda
- 3 – Exibir assentos
- 4 – Sair do Programa

Os assentos devem possuir a seguinte legenda

V – Vago / O – Ocupado

Ao vender o algoritmo deve verificar o status da cadeira para evitar vender o mesmo ingresso 2x.