

Vetores

Um vetor é uma variável composta unifilar que permite o acesso a suas posições de memória através de um índice

Os dados de um vetor serão todos do mesmo tipo conforme declaração

Em memória um vetor é armazenado sequencialmente

Assim como as variáveis, o vetor tem que ser declarado. Sua declaração é similar à declaração de variáveis, definindo primeiro o seu tipo, em seguida do seu nome e por fim a sua dimensão entre colchetes (opcional se for atribuir valores a ele na declaração)

Inteiro
vetor[5]

Índice i:
0,1,2,3,4,5

Vetores

Armazenar a nota de um aluno é possível utilizando uma variável do tipo real. Mas para armazenar as notas de todos os alunos de uma turma?

Seria necessário criar uma variável para cada aluno? E se cada turma tiver quantidade de alunos variáveis? E os nomes de cada um dos alunos?

Poderíamos armazenar estes dados em variáveis, porém o controle de muitas variáveis em um programa não é uma solução prática. Ao invés disso, utiliza-se uma estrutura de dados que agrupa todos estes valores em um nome único. Esta estrutura chama-se vetor.

Vetores

Um vetor pode ser visto como uma variável que possui diversas posições, e com isso armazenar diversos valores, porém todos do mesmo tipo.

inteiro vetor[5]

caracter vetor2[200]

//vetores inicializados

real vetor3[2] = {1.4,2.5}

logico vetor4[4] = {verdadeiro,falso,verdadeiro,verdadeiro}

cadeia vetor5[] = {"Questão","Fundamental"}

//Mudando o valor do vetor5 na posição 0 de "Questão" para "Pergunta"

vetor5[0] = "Pergunta"

Vetores

- Elementos individuais são acessados por sua posição no vetor.
- Como um vetor tem mais de uma posição, deve-se indicar qual posição do vetor se quer fazer acesso. Para isso é necessário usarmos um índice.
-
- O índice é um valor inteiro que aparece sempre **entre colchetes "[]"** após o **nome do vetor**.



```
1. programa  
2. {  
3.   funcao inicio()  
4.   {  
5.       //Declaração de um vetor de inteiros  
6.       // de cinco posições já inicializado.  
7.       inteiro vetor[5] = {15,22,8,10,11}  
8.  
9.       //Imprime o valor 15 correspondente  
10.      // ao primeiro elemento do vetor.  
11.      escreva(vetor[0])  
12.      escreva("\n")
```

```
1. //Imprime o segundo elemento do vetor
2. escreva(vetor[1])
3. escreva("\n")
4.
5. //Imprime o valor 11 correspondente
6. // ao último elemento do vetor
7. escreva(vetor[4])
8.
9. //Declaração de um vetor de reais de dez posições
10. real outro_vetor[10]
11.
12. //Declaração de um vetor de caracteres onde o tamanho
13. // é definido pela quantidade de elementos da inicialização
14. caracter nome[] = {'P','o','r','t','u','g','o','l'}
15. }
```



```

    }
}

cadeia nome
real nota[3], media

escreva("Digite seu nome: ")
leia(nome)

escreva("\n")

escreva("Digite a primeira nota: ")
leia(nota[0])

escreva("Digite a segunda nota: ")
leia(nota[1])

escreva("Digite a terceira nota: ")
leia(nota[2])

/* Calcula a média final do usuário */
media = (nota[0] + nota[1] + nota[2]) / 3

limpa()

se (media >= 6)
{
    escreva("Parabéns ", nome, "\nVocê foi aprovado com a média ", mat.arredondar(media, 2))
}
senao
{
    escreva("Que pena ", nome, "\nVocê foi reprovado com a média ", mat.arredondar(media, 2))
}

```

Como ficaria a
algoritmo de cálculo da
média usando vetores?

```

programa
{
    inclua biblioteca Matematica --> mat

    funcao inicio()
    {
        cadeia nome
        real nota[3], media=0,somanota =0

        escreva("Digite seu nome: ")
        leia(nome)

        escreva("\n")

        para (inteiro posicao = 0; posicao < 3; posicao++)
        {

            escreva("Digite a primeira nota: ")
            leia(nota[posicao])

            somanota=somanota+nota[posicao]

        }

        /* Calcula a média final do usuário */

        media = somanota / 3
    }
}

```

```

    se (media >= 6)
    {
        escreva("Parabéns ", nome, "\nVocê foi aprovado com
        média ", mat.arredondar(media, 2))
    }
    senao
    {
        escreva("Que pena ", nome, "\nVocê foi reprovado co
        média ", mat.arredondar(media, 2))
    }

    escreva("\n")
}

```


Exercicio

1- Escreva um algoritmo que solicite ao usuário a entrada de 5 números, e que exiba o somatório desses números na tela. Após exibir a soma, o programa deve mostrar também os números que o usuário digitou, um por linha.

Exercicio

2 - Escreva um algoritmo que solicite ao usuário a entrada de 5 nomes, e que exiba a lista desses nomes na tela.

Após exibir essa lista, o programa deve mostrar também os nomes na ordem inversa em que o usuário os digitou, um por linha.

Matrizes

Os vetores permitem solucionar uma série de problemas onde é necessário armazenamento de informações, porém ele possui a restrição de ser linear.

Por exemplo, imagine que queremos armazenar três notas obtidas por quatro alunos diferentes. Neste caso, existe outra estrutura mais adequada para armazenar os dados.

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 (com linha e colunas).

Matrizes

A imagem a seguir ilustra uma matriz que armazena três notas de quatro alunos:

Nova tabela

Posição	0	1	2
0	10	7	6
1	9	8	8
2	5.5	9	9



Diagram illustrating a 4x4 matrix structure with row and column indices.

LINHAS (Rows) are indicated by green arrows and numbers 0, 1, 2, 3.

COLUNAS (Columns) are indicated by purple arrows and numbers 0, 1, 2, 3.

	0	1	2	3
0	45	32	83	95
1	26	14	37	42
2	40	43	65	77
3	74	79	48	55

Matrizes

```
//Declaração de uma matriz de inteiros
// de duas linhas e duas colunas já inicializado.
    inteiro matriz[2][2] = {{15,22},{10,11}}

//Atribui -1 na primeira linha e segunda
// coluna da matriz.
matriz[0][1] = -1

//Imprime o valor 15 correspondente
// a primeira linha e primeira coluna da matriz.
inteiro i = 0
escreva(matriz[i][0])
escreva("\n")

//Imprime o valor 11 correspondente
// a última linha e última coluna da matriz.
escreva(matriz[1][1])
```

Matrizes

```
programa{
```

```
    inteiro l, c, mat[3][3] = {{45,86,95},{71,42,36},{49,16,35}}
```

```
    para(l = 0; l < 3; l++){
```

```
        para(c = 0; c < 3; c++){
```

```
            escreva(mat[l][c], " ")
```

```
        }
```

```
        escreva("\n")
```

```
    }
```

```
}
```

```
}
```

Como percorrer uma Matriz?

Percorrer uma matriz é semelhante percorrer um vetor. Contudo, devemos levar em conta as Dimensões. Um vetor possui uma dimensão, então precisará de **duas repetições**

Uma para o **índices da Linha** e uma para os **índices da coluna**.

//Escreva um algoritmo que leia os elementos de uma matriz 5x5 de inteiros. Ao final, o algoritmo deve mostrar a soma de todos os elementos.

programa

{

funcao inicio()

{

inteiro matriz[3][3] , c ,l, x = 0

escreva("\nInforme os valores da matriz: ")

para(l = 0; l < 3; l++){

para(c = 0; c < 3; c++){

leia(matriz[l][c])

x += matriz[l][c]

}

}

Continuação ..

```
escreva("\nMatriz: \n")  
para( l = 0; l < 3; l++){  
  para(c = 0; c < 3; c++){  
    escreva("[",matriz[l][c],"]")  
  }  
  escreva("\n")  
}
```

```
escreva("\nA soma dos elementos ж ",x)  
}  
}
```