UNIVALI CTTMar Kobrasol - São José - Ciência da Computação 1per

VARIÁVEIS HOMOGÊNEAS UNIDIMENSIONAIS - VETOR

Definição: corresponde a posições consecutivas de memória, identificadas por um mesmo nome, individualizadas por índices e cujo conteúdo é do mesmo tipo.

- ⇨ uma única variável capaz de armazenar um conjunto de valores.
- ⇨ o tamanho máximo do vetor depende da quantidade de memória disponível.
- ⇨ cada elemento do vetor é tratado como se fosse uma variável simples. Para referenciar um elemento do vetor utiliza-se o nome do vetor e a identificação do elemento (índice) entre colchetes.

	Índice	1	2	3	4	5	6	...	10
vetorX	Valor								

vetorX[1] <- 12 { está atribuindo o valor 12 a 1ª posição do vetorX }

	Índice	1	2	3	4	5	6	...	10
vetorX	Valor	12							

O tipo do índice do vetor pode ser inteiro, caracter, lógico (boolean) – NÃO É PERMITIDO ÍNDICE REAL OU STRING.
O valor índice de um vetor pode ser obtido como um valor constante, uma variável ou uma expressão.

Especificação de vetor em algoritmo:
nome_var: vetor [limite_inferior .. limite_superior] de tipo

O número de elementos (tamanho) do vetor é dado por (limite_superior - limite_inferior +1)
Exemplos.: idades: vetor [1..5] de inteiro
nomes: vetor [1..5] de caractere
A declaração acima corresponde à declaração de 10 variáveis:
- nomes[1], nomes[2], nomes[3], nomes[4] e nomes[5]
- idades[1], idades[2], idades[3], idades[4] e idades[5].

NA LINGUAGEM C++
- os elementos do vetor são sempre numerados por índices INTEIROS iniciados em 0 (zero).
- declaração: tipo nome_vetor [tamanho];

Ex.: float pesos[200]; unsigned int idades[5];

	Índice	0	1	2	3	4
idades	Valor					

UNIVALI CTTMar Kobrasol - São José - Ciência da Computação 1per

VARIÁVEIS HOMOGÊNEAS UNIDIMENSIONAIS - VETOR

EXEMPLO – VISUALG:

```
1. Algoritmo "vetores"
2. var
3.      numero: vetor [1..5] de inteiro  { vetor com 5 posicoes }
4.      pos: inteiro
5. inicio
6.      para pos de 1 ate 5 faca
7.              leia (numero[pos])
8.      fimpara
9.      para pos de 1 ate 5 faca
10.             escreval (numero[pos])
11.     fimpara
12. fimalgoritmo
```

EXEMPLO – PORTUGOL:

```
programa
{
        inclua biblioteca Util --> util
        funcao inicio()
        {
                inteiro vet[5]          // vetor com 5 posicoes
                // preenche o vetor
                para (inteiro pos = 0; pos < 5; pos++)
                {
                        vet[pos] = util.sorteia(1, 10)  // Sorteia um número e atribui à posição
                }
                // Exibe o vetor
                escreva ("Vetor lido:\n")
                para (inteiro pos = 0; pos < 5; pos++)
                {
                        escreva (vet [pos], " ")
                }
        }
}
```

UNIVALI CTTMar Kobrasol - São José - Ciência da Computação 1per

VARIÁVEIS HOMOGÊNEAS UNIDIMENSIONAIS - VETOR

Ordenar/**Classificar** um vetor com n elementos consiste em reorganizá-los segundo algum critério de ordenação, gerando uma sequência de valores crescente ou decrescente. Inúmeros métodos existentes.

MÉTODO DA SELEÇÃO DIRETA: Na ordenação crescente: procura o menor e coloca na 1ª posição (joga o outro valor para a posição original do menor). E assim vai fazendo com o 2º menor, 3º menor ... até o final do vetor.

ALGORITMO selecaoDireta

```
INÍCIO
{ leitura do tamanho e elementos do vetor }
{ inicio do processo de ordenação – CRESCENTE !!! }
    PARA i DE 1 ATE ( n – 1 ) FAÇA
       posicao <- i
       PARA j DE ( i + 1 ) ATE n FAÇA
          SE vetor [ j ] < vetor [ posicao ] ENTAO { ordem decrescente: usar sinal > }
               posicao <- j
          FIM SE
       FIM PARA
       auxiliar <- vetor [ posicao ]
       vetor [ posicao ] <- vetor [ i ]
       vetor [ i ] <- auxiliar
    FIM PARA
{ apresentação do vetor ordenado }
FIM
```

Pesquisar/**procurar** um elemento em um vetor com n elementos consiste em encontrá-lo (ou não) segundo algum critério. Inúmeros métodos.

MÉTODO DE PESQUISA SEQUENCIAL: Para qualquer conjunto de dados (ordenado ou não). Consiste em procurar sequencialmente um valor dentro do vetor (responde: o elemento existe ou não?).

ALGORITMO pesquisaSequencial

```
INÍCIO
{ leitura do tamanho e elementos do vetor e do elemento a ser pesquisado }
{ inicio do processo de pesquisa}
    achouElemento <- FALSO
    posicao <- 1
    ENQUANTO posicao <= tamanhoVetor E NAO(achouElemento) FACA
       SE vetor [ posicao ] = elemento ENTAO
              achouElemento <- VERDADE
       SENAO
              posicao <- posicao + 1
       FIM SE
    FIM ENQUANTO
{ apresentação da resposta se achou ou não o elemento }
FIM
```

Pergunta: oque pode ser melhorado quanto a resposta deste método ??