

Algoritmos e Programação

Estruturas de dados - Vetores

Prof. Dr. Joaquim V. C. Assunção

CENTRO DE TECNOLOGIA
UFSM
2025



Conceitos

- São variáveis compostas **homogêneas**.
- Também conhecidas como **arrays**.
- Todo conjunto é representado por um único **nome**.
- Estruturas de dados **unidimensionais (só precisam de um índice)**.

Declaração - Exemplos

```
inteiro : V[20];
```

```
inteiro : V[0:19];
```

```
inteiro : V[];
```

```
V : vetor[0..19] de inteiro;
```

```
V : vetor[0:19] de inteiro;
```

Exemplo

$V[0] \leftarrow 2;$

$V[1] \leftarrow 0;$

$V[2] \leftarrow 0;$

$V[3] \leftarrow 1;$

$V =$

2	0	0	1
---	---	---	---

Vetor de Caracteres

- Vetor de Caracteres, *literal* ou *string*.

```
string foo;
```

```
foo ← "Prof. Joaquim";
```

```
Imprima(foo)
```

Vetor de Caracteres

- Vetor de Caracteres, *literal* ou *string*.

```
string foo;
```

```
foo ← "Olá \n";
```

```
imprima(foo + "Mundo \t" + "do código");
```

Vetor de Caracteres

- Vetor de Caracteres, *literal* ou *string*.

```
string foo;
```

```
foo ← "Esta e uma grande string";
```

```
baa ← "Esta e uma pequena str..g"
```

```
strcmp(foo,baa);
```

Vetores

- Vetores de outros tipos também são possíveis.

```
Inteiro meuVet[9];
```

```
meuVet[2] ← 1;
```

```
meuVet[1] ← 2;
```

```
meuVet[3] ← 4;
```

```
meuVet[1] ← meuVet[2];
```

```
Imprimir (meuVet);
```

Operação comum

- Vetores de outros tipos também são possíveis.

```
tamanho (meuVet) ;
```

```
tamanho (foo) ;
```

```
fooLiteral.length();  
sizeof(fooNumerico) / sizeof(fooNumerico[0]);
```

Conceitos

- São variáveis compostas **homogêneas**.
- Também conhecidas como **arrays**.
- Todo conjunto é representado por um único **nome**.
- Estruturas de dados **unidimensionais (só precisam de um índice)**.

Exercício

1) Criar um algoritmo que entre com uma palavra e imprima o inverso, conforme o exemplo a seguir:

- Entrada: UFSM
- Saída: MSFU

Solução em C++

...

```
string palavra;
```

```
    cout << "Digite uma palavra\n";
```

```
    cin >> palavra;
```

```
    for (int i = palavra.length()-1; i >=0 ; i--){  
        cout << palavra[i];
```

```
    }
```

Exercícios

2) Entrar com uma mensagem e imprimir quantas letras A, E, I, O e U (cada) tem esta mensagem.

3) Entrar com uma mensagem e criptografá-la da seguinte maneira:

A – X; E – Y; I – W; O – K; U – Z

Exercício

4) Criar um algoritmo que entre com uma palavra e imprima conforme o exemplo a seguir:

- Entrada: CASA

- Saída:

C

CA

CAS

CASA

Notas Legais

O conteúdo é baseado na terceira edição do livro “Lógica de Programação” de André Forbellone e Henri Eberspacher (ver bibliografia da disciplina).