



# Universidade Estadual de Mato Grosso do Sul

## Ciência da Computação

### Algoritmos e Estruturas de Dados I (AED-I)

## Array

**Prof. Nilton**

[nilton@comp.uems.br](mailto:nilton@comp.uems.br)

# Array

- Coleções de um mesmo tipo em sequência;
- Podem ser de tipos primitivos (*char*, *int*, *float*, *double*), de um outro array, de uma estrutura, de enumeração...
- Pode se ter um array de inteiros ou um array de caracteres ou um array de arranjo de pontos flutuantes
  - Mas, não se pode ter um array que contenha inteiros e pontos flutuantes

# Array

- **Declarando um Array**

- `int notas[5];`            `/* Array de 5 inteiros */`
- `char letras[5];`        `/* Array de 5 caracteres */`

- **Declarando e inicializando um Array**

- Como uma variável comum, um array não é inicializado automaticamente
- `int notas[5] = {4, 6, 6, 9, 8};`
- `char letras[5] = {65, 'E', 'I', 'O', 'U'};`
- `char nome[100] = "ANA";`

# Array

- **Um array de tamanho  $n$  tem suas posições indexadas de 0 a  $n-1$** 
  - `int notas[5] = {3, 7, 8, 9, 2};`
  - `notas[3] = 7;`
  - `printf("%d",notas[4]);`
- **Para obter o tamanho de um array, basta dividir o seu tamanho pelo tamanho ocupado por cada elemento utilizando a função *sizeof*. Exemplo:**
  - `int tam = sizeof(notas) / sizeof(int);`

# Declarando, inicializando e iterando um array de inteiros

```
int main()
{
    int notas[10];
    int tam = sizeof(notas) / sizeof(int);
    int i;

    for(i = 0; i < tam; i++)
        notas[i] = 0;

    return 0;
}
```

---

# Declarando, inicializando e iterando um array de caracteres

```
int main()
{
    char vogais[5] = {'a', 'e', 'i', 'o', 'u'};
    int tam = sizeof(vogais) / sizeof(char);
    int i;

    for(i = 0; i < tam; i++)
        printf("%c ", vogais[i]);

    return 0;
}
```

# Array - acesso aos elementos

- **Array permite acessar ou alterar qualquer um de seus elementos**
- **Array em C sempre inicia na posição ZERO e seu último elemento está na posição inferior ao tamanho (n-1)**

```
char letras[10];  
letras[0] = 'A';  
letras[9] = 'H';  
/* letras[10] = 'I'; --> ERRO */
```

# Array de caracteres como uma string

- Em C, não existe o tipo string. Ele é simulado usando um array de caracteres em que `'\0'` delimita seu final
- Suas posições válidas são de 0 a  $n-2$
- A posição  $n-1$  deve ter o terminador nulo
- Seu tamanho é calculado pelo uso da função *strlen* da biblioteca *string.h*
- A leitura utiliza a função *gets* e a impressão usa-se *%s*



# Array de caracteres como uma string

```
#include <stdio.h>
#include <string.h>

int main() {
    char nome1[50];
    char nome2[12]="Mauro Silva";
    char nome3[] = "Mauro Silva";
    char nome4[12]={ 'M','a','u','r','o',' ','S','i','l','v','a','\0'};
    char nome5[]={ 'M','a','u','r','o',' ','S','i','l','v','a','\0'};

    printf("\nDigite seu nome completo: ");
    gets(nome1);
    printf("\n%s tem %d caracteres\n",nome1,strlen(nome1));

    printf("\nDigite seu nome completo: ");
    scanf("%s",nome1);
    printf("\n%s tem %d caracteres\n",nome1,strlen(nome1));
    getch();

    printf("\n%s tem %d caracteres\n",nome2,strlen(nome2));
    printf("\n%s tem %d caracteres\n",nome3,strlen(nome3));
    printf("\n%s tem %d caracteres\n",nome4,strlen(nome4));
    printf("\n%s tem %d caracteres\n",nome5,strlen(nome5));
    return 0;
}
```

# Copiando um array

- **Observe**

```
int vA[4] = {9, 2, 5, 20};  
int vB[4];
```

- **Cópia incorreta**

```
vB = vA;
```

- **Cópia correta**

```
for(i=0; i<4; i++)  
    vB[i] = vA[i];
```