

# AULA 07

{introcomp}

VETORES II

# Vetores II

---

## **Strings**

Leitura e exibição

Biblioteca string.h

Operações com Strings

## **Matrizes**

Definição de Acesso

Operações com Matrizes



# Vetores II

---

## Objetivo

Compreender a definição e dominar a implementação com strings e matrizes em C.



# Vetores II

## Strings

Em C, uma string, ou cadeia, é uma sequência de caracteres finalizada por um byte nulo, representado por `'\0'`.



# Vetores II

## Strings

```
char s[6];  
s[0] = 'w';  
s[1] = 'o';  
s[2] = 'r';  
s[3] = 'l';  
s[4] = 'd';  
s[5] = '\0';
```



# Vetores II

## Strings

0	1	2	3	4	5
'w'	'o'	'r'	'l'	'd'	'\0'



# Vetores II

## Strings

```
char s[6];  
s[0] = 'w';  
s[1] = 'o';  
s[2] = 'r';  
s[3] = 'l';  
s[4] = '\0';  
s[5] = '\0';
```



# Vetores II

## Strings

0	1	2	3	4	5
'w'	'o'	'r'	'l'	'\0'	'\0'



# Vetores II

## Strings

```
char s[6];  
s[0] = 'w';  
s[1] = 'o';  
s[2] = '\0';  
s[3] = 'r';  
s[4] = 'l';  
s[5] = 'd';
```



# Vetores II

## Strings

0	1	2	3	4	5
'w'	'o'	'\0'	'r'	'l'	'd'



# Vetores II

## Inicializando strings

```
char nome[10] = {'P','e','d','r','o'};  
//Aloca 10 espaços para string
```

```
char nome[10] = "Pedro";  
//Aloca 10 espaços para string
```

```
char nome[] = "Pedro";  
//Calcula a quantidade automaticamente e aloca 6  
espaços ("Pedro" + '\0')
```



# Vetores II

## Leitura e exibição de strings

Para ler e exibir strings, podemos usar `scanf()` e `printf()` com o formato `%s`.

```
char nome[100];  
printf ("Entre com seu nome:\n");  
scanf ("%s", nome); // não se usa o &  
printf ("Seu nome: %s\n", nome);
```



# Vetores II

## Leitura e exibição de strings

O scanf, por padrão, lê caractere por caractere até que se encontre um espaço (' ') ou uma quebra de linha ('\n').

**Como ler uma string com espaços e até mesmo com quebras de linha?**



# Vetores II

## Leitura e exibição de strings

**Resposta:** utilizando-se do especificador de extensão ‘[<Intervalo>]’.

### **Leia caracteres enquanto:**

- Não forem uma quebra de linha: `%[^\n]`
- Forem letras do alfabeto:

`%[abcdefghijklmnopqrstuvwxyz] ou %[a-z]`

- O especificador acima só lê enquanto forem letras minúsculas.

Para ler as maiúsculas também, podemos fazer: `%[A-Za-z]`



# Vetores II

```
char nome[100];  
printf ("Entre com seu nome completo:\n");  
scanf ("%[^\\n]", nome);  
printf ("Seu nome completo: %s\\n", nome);
```



# Vetores II

**Como realizar a leitura corretamente?  
Exemplo do working 6 - praticando 3:**

Entrada:

6

aaooiaa



# Vetores II

**Como realizar a leitura corretamente?**

**Exemplo do working 6 - praticando 3:**

Entrada:

6

aaoniaa

Entre o número 6 e o primeiro caractere 'a' existe um '\n' que precisa ser "descartado" para a leitura ser feita corretamente. Como fazer isso?



# Vetores II

## Descartando um caractere:

Entrada:

6

aaoniaa

Podemos descartar o '\n' utilizando a seguinte expressão: `scanf("%*c");`



# Vetores II

## Descartando um caractere:

Entrada:

6

aaioiaa

Portanto, a leitura correta seria:

```
scanf("%d",&num) //armazena o 6 na variável num
```

```
scanf("%*c"); //descarta o '\n'
```

```
//...fazer a leitura dos caracteres aaioiaa normalmente.
```



# Vetores II

---

## Exemplo 1

**Faça um programa que leia uma frase até o '\n' e imprima na tela.**



# Vetores II

## Funções para manipulação de strings

- Calcular o tamanho de uma string;
- Comparar se duas strings são iguais, maior ou menor;
- Copiar uma string em outra;
- Concatenar uma string com outra;

Para utilizar essas funções, devemos incluir a biblioteca `string.h`:

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



# Vetores II

Para descobrir o tamanho de uma string (sem o '\0'):

**strlen** (char str[])

- Retorna o tamanho (int).

Exemplo:

```
char nome[] = "Lucio";
```

```
int tamanho = strlen(nome);
```

```
printf("%d",tamanho); //será exibido 5 na tela.
```



# Vetores II

Para copiar um string para outra:

```
strcpy (char destino[], char fonte[]);
```

```
strncpy (char destino[], char fonte[], int qtdChar);
```

Exemplo:

```
char nome[10], nome2[10];
```

```
strcpy(nome, "Lucas");
```

```
strncpy(nome2, "Isabel Maria",4);
```

```
printf("%s",nome) //Será exibido "Lucas"
```

```
printf("%s",nome2) //Será exibido "Isab"
```



# Vetores II

Para comparar duas strings:

**strcmp** (char str1[], char str2[]);

Retorna:

0, quando iguais

> 0, quando str1 maior

< 0, quando str1 menor

**strncmp** (char str1[], char str2[], int qtdChar);

Exemplo:

```
char nome1[10], nome2[10];
```

```
scanf("%s", nome1);
```

```
scanf("%s", nome2);
```

```
if(strcmp(nome1, nome2) == 0) printf("As strings são iguais!\n");
```

```
else printf("As strings são diferentes.\n");
```



# Vetores II

Para concatenar (“somar”) duas strings:

```
strcat (char destino[], char fonte[]);
```

```
strncat(char destino[], char fonte[], int qtdChar);
```

Exemplo:

```
char nome[50] = “João”;
```

```
strcat(nome, “ Vitor ”);
```

```
printf(“%s\n”, nome); //Será exibido “João Vitor ”.
```

```
strncat(nome, “Loureiro”, 3);
```

```
printf(“%s\n”, nome); //Será exibido “João Vitor Lou”.
```



# Vetores II

## Exemplo 2

**Faça um programa que leia uma frase até o '\n', transforme todos os caracteres da string em maiúsculos e imprima na tela.**

<b>Entrada</b>	<b>Saída</b>
oi galera	OI GALERA



# Vetores II

## Exemplo 3

**Faça um programa que leia duas strings e exiba as duas concatenadas.**

<b>Entrada</b>	<b>Saída</b>
Ola Mundo!	Ola Mundo!



# Vetores II

## Exemplo 4

**Crie um programa que dado um conjunto de strings imprima a maior e menor string.**

<b>Entrada</b>	<b>Saída</b>
3 Abc Ace Ucg	Maior: Ucg Menor: Abc



# Vetores II

## Matrizes

Matrizes são vetores cujos itens também são vetores.

	0	1	2	3	4
0					
1					
2					
3					



# Vetores II

## Matrizes - Declaração

<tipo> <nome\_da\_matriz>[<qtdlinhas>][<qtdcolunas>;

Exemplo: **int** matriz[4][5];

	0	1	2	3	4
0					
1					
2					
3					



# Vetores II

## Matrizes – Atribuição (int)

Atribuição através dos índices:

```
int matriz[4][5]; //declaração de uma matriz com 4 linhas e 5 colunas.
```

```
matriz[2][1] = 45; //atribuição no segundo elemento da terceira linha.
```

	0	1	2	3	4
0					
1					
2		45			
3					



# Vetores II

## Matrizes – Atribuição (int)

`matriz[3][2] = 20; //atribuição no terceiro elemento da quarta linha.`

	0	1	2	3	4
0					
1					
2		45			
3			20		



# Vetores II

## Declarando e percorrendo uma matriz:

```
int l, c, i, j;
scanf("%d %d", &l, &c);
int matriz[l][c];

for (i=0; i<l; i++)
    for (j=0; j<c; j++)
        scanf("%d", &matriz[i][j]);
```



# Vetores II

## Exemplo 5

**Faça um programa que dadas as dimensões M e N de uma matriz, leia e imprima MxN valores e os imprima em formato de matriz.**

<b>Entrada</b>	<b>Saída</b>
3 3	1 2 3
1 2 3	4 5 6
4 5 6	7 8 9
7 8 9	



# Vetores II

## Exemplo 6

**Faça um programa que compare duas matrizes 3x3, e diga se elas são iguais.**

	<b>Entrada</b>	<b>Saída</b>
Matriz 1	1 1 1 2 2 2 3 3 3	Iguais
Matriz 2	1 1 1 2 2 2 3 3 3	



# Vetores II

## Matrizes de char

Se comportam como vetores de string.

Exemplo: `char matriz[4][5];`

	0	1	2	3	4
0	'L'	'a'	'r'	'a'	'\0'
1	'J'	'o'	'a'	'o'	'\0'
2	'H'	'u'	'g'	'o'	'\0'
3	'E'	'n'	'z'	'o'	'\0'



# Vetores II

## Lendo uma matriz de char (como vetor de strings):

OBS: Cada string tem no máximo 20 caracteres nesse exemplo

```
int n, i;  
scanf ("%d", &n);  
char matriz[n][21];  
  
for (i=0; i<n; i++)  
    scanf ("%s", matriz[i]);
```



# Vetores II

## Exemplo 7

**Faça um programa que dado N nomes, imprima-os. Os nomes possuem no máximo 15 caracteres.**

<b>Entrada</b>	<b>Saída</b>
3 Pedro Anselmo Lucio Sandrini Lucca	Pedro Anselmo Lucio Sandrini Lucca



# Vetores II

## Matrizes – Passagem como parâmetro de função

```
void func1 (int mat[][10]) {  
    .  
    .  
    .  
}  
  
int main() {  
    int mat[20][10];  
    func1 (mat);  
    return 0;  
}
```



# Vetores II

## Exemplo 8

**Faça um programa que dado N nomes, imprima-os em ordem alfabética. Os nomes podem ter até 20 caracteres.**

**Faça uma função para a ordenação.**

<b>Entrada</b>	<b>Saída</b>
3 Pedro Anselmo Lucio Sandrini Lucca	Lucca Lucio Sandrini Pedro Anselmo



# AULA 07

{introcomp}

VETORES II