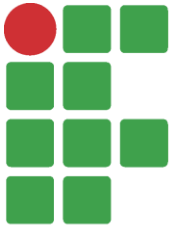


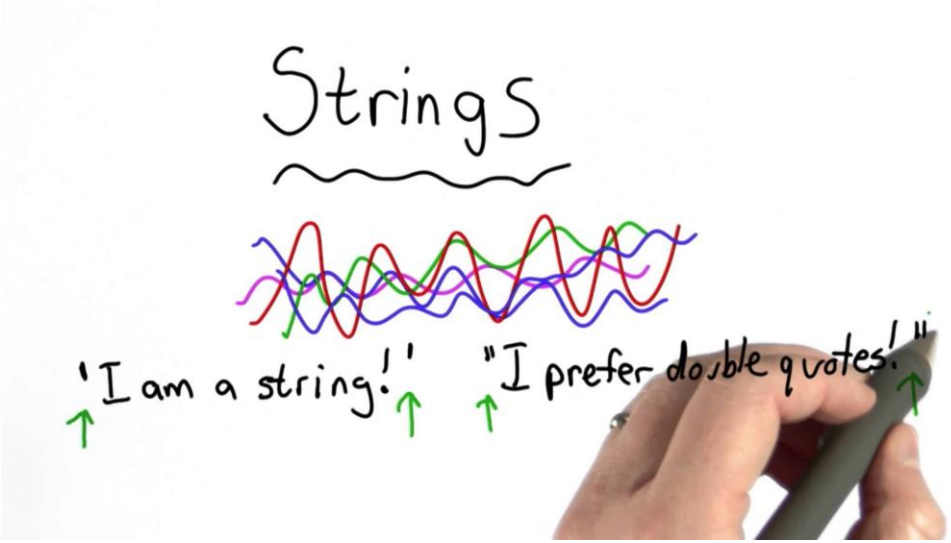
INSTITUTO FEDERAL
Triângulo Mineiro | Campus Patrocínio

LINGUAGENS DE PROGRAMAÇÃO

GILBERTO VIANA DE OLIVEIRA

Nesta aula...

- Strings
  - Manipulação de strings



Introdução

- ***String*** é o nome que usamos para definir uma sequência de caracteres adjacentes na memória do computador.
- Esse conjunto de caracteres pode ser:
  - Uma palavra;
  - Uma frase;
- É formado por um vetor do tipo char.
  - As regras para declaração são as mesmas regras de declaração de vetores.

Introdução

```
char str[6];
```

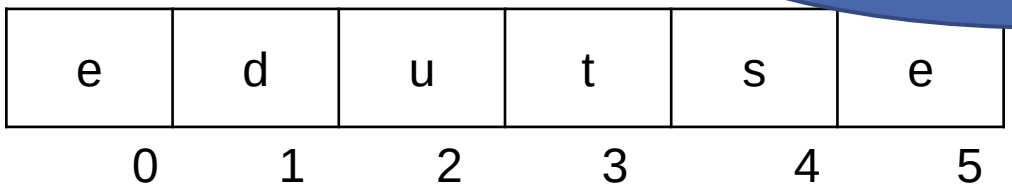
- A declaração acima cria na memória do computador uma string (array de caracteres), de nome **str** e tamanho igual a 6.
- No entanto, por ser uma string, devemos ficar atentos ao fato de que as strings tem no elemento seguinte a última letra armazenado um caractere **'\0'**.
  - Esse caractere é necessário pois podemos armazenar uma palavra menor do que o tamanho definido inicialmente.

Introdução

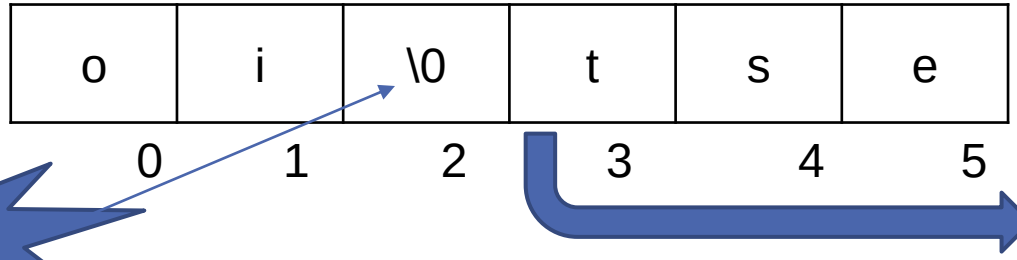
- Exemplo:

```
char str[6];
```

Ao definirmos um vetor, não sabemos seu conteúdo




```
//leia uma string e armazene em str
//digitado "oi"
```



Término da string

Posições não utilizadas

Declarando uma string

- Considerando que devemos armazenar o caractere nulo (\0), sempre devemos declarar nossa string acrescentando 1 ao tamanho desejado.
- Exemplo:
  - Criar uma string para armazenar a sigla de um estado brasileiro.

```
char estado[3];
```

Declarando uma string

- Modos para declarar e inicializar uma string:

```
//declarando e inicializando
char texto[] = {'a', 'b', 'c', '\0'};
char texto2[4] = "abcd";
char texto3[] = "abcd";
```

- A atribuição direta só pode ser feita durante a declaração. Logo, o trecho abaixo está errado!

```
texto = "def";
```

Formas de ler uma String

- Podemos fazer a leitura de uma string de várias formas.
  - Ler uma string até um **espaço** ou **salto de linha**.
  - Note que não utilizamos o &.
  - Só lê uma palavra.

```
char texto[10];
scanf("%s", texto);
```

- Observação: Os caracteres aceitos em linguagem C respeitam a tabela ASCII;
  - https://pt.wikipedia.org/wiki/ASCII

Formas de ler uma String

- Podemos resolver o problema de armazenarmos apenas a primeira palavra com o scanf().
- No exemplo abaixo, o scanf() irá ler até encontrar um caractere de quebra de linha (\n).

```
char texto[10];
scanf("%[^\n]", texto);
```

Formas de ler uma String

- Outra forma de ler uma string é usar a função gets (stdio.h).
- Essa função lê uma cadeia de caracteres até encontrar um “enter”.
- É indicado para ler frases e nomes completos.

```
char texto[10];
gets(texto);
```

fflush(stdin)

- Assim como foi discutido anteriormente na disciplina, a leitura de caracteres pode causar problemas, pois ao apertarmos a tecla ENTER, ele pode ficar gravado no buffer de entrada como um caractere \n.
- Isso pode servir de entrada para a próxima leitura que acontecerá no código.
- Para evitar esse problema, utilizar a função **fflush(stdin)**, que limpa o buffer de entrada.

Escrevendo uma string na tela

- Para escrevermos uma string na tela, podemos utilizar:

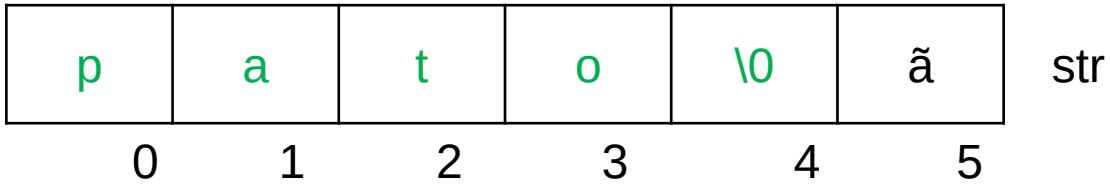
```
//imprimindo strings
printf("%s", texto);
puts(texto);
```

Escrevendo uma string na tela

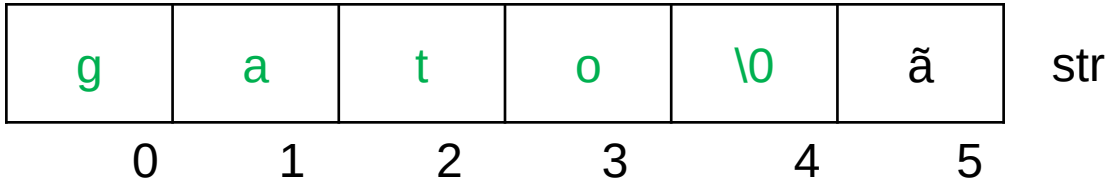
- Se preferir, poderá imprimir caractere por caractere, usando um laço de repetição.
- Porém esse laço não deve terminar apenas quando chega ao tamanho do array criado.
- O laço deve terminar ao encontrar o caractere ‘\0’;

Observações sobre strings:

- Ao trabalharmos com strings, podemos acessar seus elementos de forma individual, assim como fazíamos com arrays.

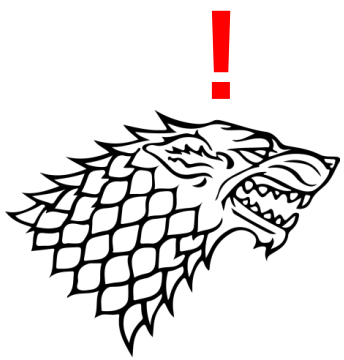



```
str[0] = 'g';
```



Observações sobre strings:

- Não é possível atribuir valores diretos à strings após sua criação. Isso também se aplica caso deseja-se atribuir um valor de uma string para outra.
- Em ambos os casos. A operação está errada!

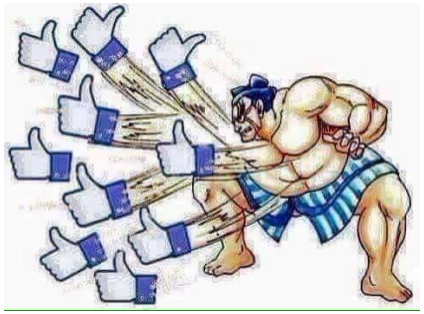


```
char texto[20], texto2[20];

texto = "winter is coming";
texto2 = texto;
```

Observações sobre strings:

- Esse erro acontece porque a linguagem C não aceita a atribuição direta de um array para outro.
- Lembre-se: Uma string é um array!
- Nesse caso, uma forma de copiar uma string1 para uma string2, é copiar caractere por caractere de um array para o outro.
  - Usando um laço de repetição!



Exercício

1. Declare duas strings com capacidade para 15 caracteres. Leia a entrada para a primeira string. Copie o texto da primeira string para a segunda. Ao final, imprima as duas strings.
2. Escreva um programa que lê uma string de no máximo tamanho 10. Ao final, a string deve imprimir o texto armazenado e mostrar o tamanho do texto digitado.
3. Faça um programa que lê uma string (tamanho 10) e imprima o texto armazenado na ordem inversa.

Exercício

- Questão 2 – Uma forma de resolver

```
int main(){
    char a;
    int tam = 0;
    char texto[11];
    gets(texto); //le a string

    a = texto[0]; //inicializa os caracteres lidos

    while(a != '\0'){
        tam++;
        a = texto[tam];
    }
    printf("%s", texto);
    printf("\nTAMANHO: %d", tam);
return 0;
}
```

Funções para manipulação de strings

- A biblioteca padrão da linguagem C possui funções desenvolvidas para a manipulação de strings na biblioteca <string.h>
- A seguir, as funções mais utilizadas serão apresentadas.
- Essa biblioteca possui diversas funções, para verificar a documentação completa da biblioteca acesse: https://www.tutorialspoint.com/c_standard_library/string_h.htm

Funções para manipulação de strings

- Determinar o tamanho de uma string.
- Função strlen() retorna o número de caracteres até o caractere ‘\0’.

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

int main(){

    char str[10] = "teste";
    int tamanho;
    tamanho = strlen(str);

    printf("%d", tamanho);

return 0;
}
```

Funções para manipulação de strings

- Copiar uma string:
  - Como visto, a linguagem C não suporta uma atribuição de um array para outro.
  - Logo, devemos fazer uma cópia, elemento ou elemento.
- Função strcpy(char *destino, char *origem)
  - Copia a sequencia de uma string origem para uma string destino.
  - A ordem mostrada deve ser respeitada.

Funções para manipulação de strings

- Exemplo:
  - Copia a palavra “teste” armazenada em str para str2;

```
int main(){

    char str[10] = "teste";
    char str2[10];

    strcpy(str2, str);

    printf("%s", str2);

return 0;
}
```

Funções para manipulação de strings

- Concatenando string
  - A operação de concatenação é outra tarefa comum ao se trabalhar com strings.
  - Basicamente, ela copia uma string para o final de outra string.
  - A forma da função é strcat (char *destino, char *origem)
  - O primeiro caractere da string origem é colocado no lugar do caractere ‘\0’ da string destino.

Funções para manipulação de strings

- Exemplo:
  - Copia a string texto para o final da string texto2, iniciando no caractere '\0' de texto.

```
int main(){

    char texto[10] = "Forever";
    char texto2[10] = "Wakanda";

    strcat(texto2, texto);

    printf("%s", texto2);

return 0;
}
```

Funções para manipulação de strings

- Comparando duas strings:
- Da mesma forma que o operador de atribuição não funciona para strings, o mesmo ocorre com operadores relacionais.
- Logo, precisamos de uma função para verificar se duas strings são iguais ou não.
- A função para determinarmos isso é:
  - int strcmp(char *texto1, char *texto2)

Funções para manipulação de strings

- Comparando duas strings:
  - Por ser uma função do tipo int, ela retornará um resultado do tipo inteiro.
  - Após uma comparação, se um valor inteiro zero for retornado, as duas strings são iguais.
  - Se um valor diferente de zero for retornado, as strings são diferentes.
  - **Observação**: a função strcmp é case sensitive, ou seja, uma comparação entre “abc” e “Abc” retornará que as strings são diferentes!!

Funções para manipulação de strings

- Comparando duas strings:
  - Armazena em compara um inteiro (zero se forem iguais)

```
int main(){

    char texto[10] = "iPhone";
    char texto2[10] = "tPhone";
    int compara;
    compara = strcmp(texto, texto2);

    if(compara == 0){
        printf("Strings sao iguais");
    }else{
        printf("Strings sao diferentes");
    }

return 0;
}
```

Outras funções

- strupr(char *str) – converte uma string para maiúscula.
- strlwr(char *str) – converte uma string para minúsculas.
- strrev (char *str) – inverte o conteúdo de uma string.
- Observações:
  - Operador de atribuição e de comparação não podem ser usados diretamente em strings (= e ==)
  - É possível manipular elementos individuais de uma string (caracteres).

Exercício

1. Faça um programa que receba uma string e imprima quantas vogais de cada ele possui.
2. Faça um programa que receba duas strings e escreva-as em ordem alfabética.
   - Para isso, é necessário estudar a documentação da função strcmp: https://tentandoblogar.wordpress.com/2009/03/15/comparando-palavras-em-c-a-funcao-strcmp/
   - Escreva em formato de comentário no seu código, como conseguiu resolver esse exercício.
3. Faça um programa que receba duas strings. Concatene a segunda string no final da primeira. Em seguida, substitua todas as letras ‘a’ existentes pela letra ‘e’. Imprima a string resultante.

Contato

Horário de atendimento disponível no portal.

E-mail do professor: gilbertooliveira@iftm.edu.br

MAY THE FORCE

BE WITH YOU

Referências

- Agradecimentos ao professor Marques Sousa por ceder parte do material utilizado na aula.
- SCHILDT, H. CC: completo e total. São Paulo: Makron Books, 2000.
- BACKES, A. Linguagem C descomplicada.
- Imagens retiradas do *google images*.