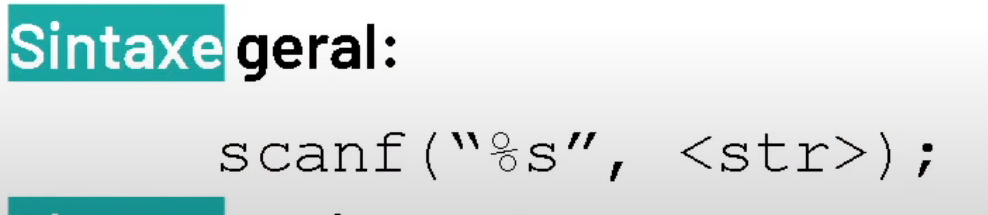
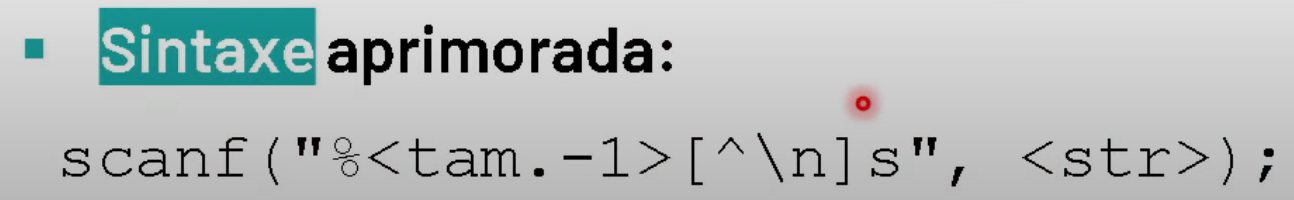
**STRINGS**

* Vetor de caracteres
* A última posição do vetor tem que ter o “\0”. Então se tenho uma palavra com 10 letras, o tamanho do vetor terá que ser 11
* Posso ler o vetor/ string com scanf
  + A sintaxe geral não considera os espaços



* Sintaxe aprimorada: consegue ler os espaços

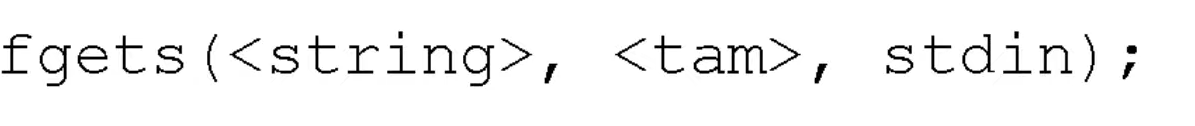


**FUNÇÃO gets()**

* Função para entrada de dados
* Não controla limite de tamanho de vetor
* Sintaxe: Gets(string)
  + gets(s)

**FUNÇÃO fgets()**

* Função para entrada de dados
* Substitui o scanf
* Sintaxe:



* Ex.: fgets(s, 10, stdin);

**FUNÇÃO puts()**

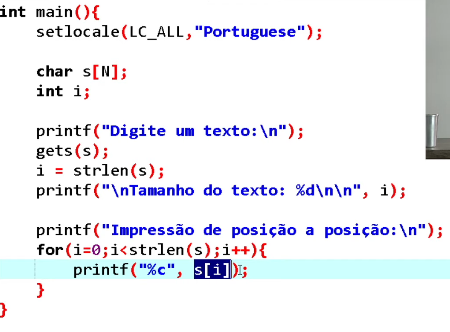
* Função para saída de dados (tipo printf)



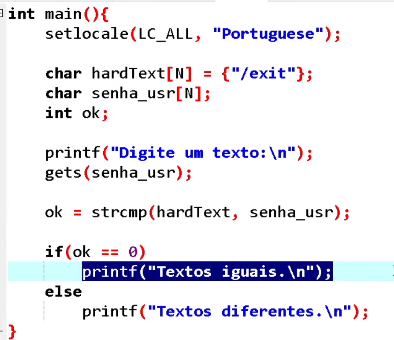
* *Função fflush(stdin):*
  + Chamar depois de entrada de dados
  + Não da problema na próxima entrada de dados
  + Limpa as sujeiras!!

**BIBLIOTECA string.h: funções importantes!**

* strcpy (<destino>; <origem>);
  + Destino: nome da string que vai receber o conteúdo
  + Origem: nome da string que vai ceder o conteúdo
  + Atribuição de variáveis simples é i=5 (por exemplo). Com string é diferente
  + Faz atribuições a strings
  + Alterar o conteúdo de uma string
* strcat (<destino>; <origem>);
  + Quando quero/juntar colar duas strings
* strlen(<string>)
  + Mostra o tamanho da string
  + Exemplo: o usuário digita algum conteudo. Quer saber o tamanho desse conteudo e imprimir cada posição



* strcomp(<string1>, <string2>)
  + Compara a igualdade entre strings
  + Se for igual produz o valor zero
  + Exemplo de senha



if (strcmp(nome, outro)==0){

printf ("\nSao iguais");

}else{

printf ("\nSao diferentes.");

}

**BIBLIOTECA locale.h: funções importantes!**

* Permite usar assentos nos programas
* Invocar dentro do main

