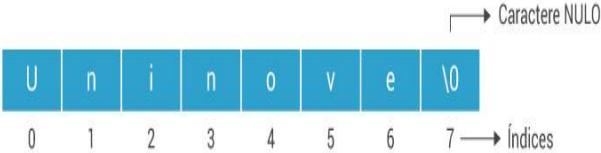


STRINGS	STRINGS
O que é a <b>estrutura de dados String</b> ?	No cotidiano sempre nos deparamos com frases, textos e palavras, os quais são representados por seqüências (ou cadeias) de caracteres. Como exemplos, podemos citar o envio de mensagens por correios eletrônicos, os editores de textos e os programas de cadastro, onde os dados são representados textualmente. Dentro do contexto de linguagens de programação, seqüências de caracteres são denominadas strings.
Como podemos <b>definir</b> uma String?	A maioria das linguagens de programação permite a representação de uma string como uma de cadeias de caracteres e inclusive possuem um tipo de dados específico chamado string para tal finalidade. Com exceção da Linguagem C não possui este tipo de dados e, por este motivo, é utilizado o conceito de vetores, os quais permitem agupar, em uma só variável, vários caracteres. Por exemplo, na Linguagem C, strings são representadas por vetores de caracteres e devem ser terminadas, obrigatoriamente, pelo caractere nulo ( \0 ). Nesse caso, devemos reservar uma adicional para o caractere, que indica o fim de cadeia.
Ilustre como uma String pode ser composta na Linguagem C...	
Como podemos <b>acessar os caracteres</b> de uma String?	Os caracteres de uma string podem ser acessados de forma individualizada. Já que a string é um vetor. Um caractere é referenciado pelo nome da string seguido do índice onde ele está armazenado entre colchetes. No exemplo abaixo referencia o terceiro caractere da string denominada palavra. O terceiro caractere, que no exemplo anterior, é a letra “i”, é referenciado pelo índice 2, já que o primeiro elemento tem índice 0.  <b>palavra [2] = ‘i’;</b>
Como podemos fazer a <b>declaração</b> de uma String?	Como uma variável simples, uma string deve ser declarada antes de sua utilização. A declaração de uma string é feita indicando o tipo de dado como "string" ou "char" (na linguagem C), o seu nome e o seu tamanho (na linguagem C). A forma geral de declaração de uma string em linguagem C é:  <b>char nome [tamanho];</b>  Onde, nome é o nome pelo qual a string será referenciada e tamanho é a quantidade de caracteres que a string pode conter. Por exemplo, veja declaração de uma string denominada palavra de 8 posições de caracteres.  <b>char palavra [8];</b>  Assim como uma variável simples, a string pode ser declarada e inicializada no início de um programa. O trecho de código a seguir mostra a inicialização da string sob 2 formas.  <b>char palavra [8] = { ‘U’ , ‘n’ , ‘i’ , ‘n’ , ‘o’ , ‘v’ , ‘e’ , ‘\0’};</b>  ou  <b>char palavra [8] = “Uninove”;</b>  <b>DICA IMPORTANTE: O conteúdo das variáveis do tipo char deve ser referenciado entre apóstrofes ( ‘ ’ ), mas o conteúdo das strings deve ser referenciado entre aspas ( “ ” ).</b>
- reservada para questão acima -	
- reservada para questão acima -	
Como podemos <b>fazer a leitura do conteúdo</b> de uma String?	Imagine que temos uma string denominada frase, que pode armazenar, no máximo, 50 caracteres. Queremos solicitar ao usuário que informe os dados para preencher esse vetor de caracteres. O trecho de código a seguir lê os dados utilizando-se a função scanf: <b>printf (“Digite a String=”);</b>  Para exibir o conteúdo de uma string, podemos utilizar a função printf, juntamente com o especificador de formato %s. O trecho de código a seguir mostra a exibição da String utilizando a função printf. <b>printf (“%s \n”, frase);</b>  Podemos também especificar um número máximo de caracteres suportados numa strings até que o caractere de mudança de linha (‘\n’) seja encontrado, o qual está representado no trecho de código a seguir: <b>printf (“Digite a String=”);</b> <b>scanf (“ %49[^\n]”, frase);</b>  Entretanto, a leitura feita com esta função não é muito utilizada, pois seu uso é muito limitado.
- reservada para questão acima -	
O que são as <b>Bibliotecas da Linguagem C</b> e que <b>funcionalidades elas trazem</b> para o uso de Strings?	Um dos recursos adicionais da linguagem C é a possibilidade de declararmos bibliotecas, cuja principal vantagem é a utilização de funções pré-definidas. Isso deixa o código mais claro e de fácil manutenção. Para utilizarmos as funções que manipulam strings, temos que declarar, no início do programa, a biblioteca “string.h”. Dentre as funções mais usadas, destacam-se: gets, puts, strcmp, strcpy e strlen.

MATRIZES		MATRIZES
Como podemos utilizar a <b>função "gets"</b> nas nossas Strings?		<p><b>gets:</b> utilizada para a leitura de uma string via teclado. Esta é a função mais indicada e utilizada para a leitura, pois não tem as limitações da função scanf. O trecho de código abaixo mostra a utilização da função gets. A sintaxe da função é a seguinte:</p> <pre>#include &lt;string.h&gt;</pre> <pre>char str [40]; printf ("\n Digite a String desejada= "); gets (str);</pre>
Como podemos utilizar a <b>função "puts"</b> nas nossas Strings?		<p><b>puts:</b> utilizada para a exibição de uma string. O trecho de código abaixo mostra a utilização da função puts. A sintaxe da função é a seguinte:</p> <pre>#include &lt;string.h&gt;</pre> <pre>char str [40]; printf ("\n Digite a String desejada= "); gets (str); puts (str);</pre>
Como podemos utilizar a <b>função "strcpy"</b> nas nossas Strings?		<p><b>strcpy:</b> utilizada para copiar o conteúdo de uma string origem para uma string destino. A primeira string terá o mesmo valor da segunda string. Podemos, também, colocar uma string qualquer entre aspas ao invés de uma variável no lugar da segunda string (origem). Importante lembrar que as duas strings devem ter o mesmo tamanho definido, sendo que a segunda string (origem) pode ser menor, nunca maior que a primeira (destino). O trecho de código abaixo mostra a utilização da função strcpy. A sintaxe da função é a seguinte:</p> <pre>#include &lt;string.h&gt;</pre> <pre>char strdest [40], strorig [40]; printf ("\n Digite a String Origem= "); gets (strorig); strcpy(strdest, strorig);</pre>
- reservada para questão acima -		<p>strcpy: utilizada para fazer comparações entre o conteúdo de 2 Strings, se são iguais no conteúdo. Nesse caso, a função retorna o valor 0 (zero) se as duas cadeias forem idênticas, um valor menor que zero se a primeira string for alfabeticamente menor que segunda string ou um valor maior que zero se a primeira string for alfabeticamente maior que a segunda string. Esta função diferencia maiúsculas de minúsculas. Podemos utilizar essa função em conjunto com a função "if" para comparar os valores das strings afim de mostrar se uma string é igual a outra ou não. O trecho de código abaixo mostra a utilização da função strcmp. A sintaxe da função é a seguinte:</p> <pre>#include &lt;string.h&gt;</pre> <pre>char str1[50], str2[50]; strcpy(str1, "Linguagem C"); printf ("\n Digite a Segunda String="); gets (str2); if ( strcmp (str1, str2) == 0 ) {     printf ("\n As Strings sao Iguais");}else { printf ("\n As Strings sao Diferentes");}</pre>
Como podemos utilizar a <b>função "strlen"</b> nas nossas Strings?		<p><b>strlen:</b> utilizada para retornar o tamanho (quantidade de letras) de uma string, desprezando o caractere nulo final (\0). Ela retorna o valor exato de caracteres. A sintaxe da função é a seguinte:</p> <pre>#include &lt;string.h&gt; char str [50]; int tam; strcpy(str, "UNINOVE"); tam = strlen(str); printf("\n A quantidade de caracteres da palavra e= %d", tam);</pre>