



ALGORITMOS E PROGRAMAÇÃO

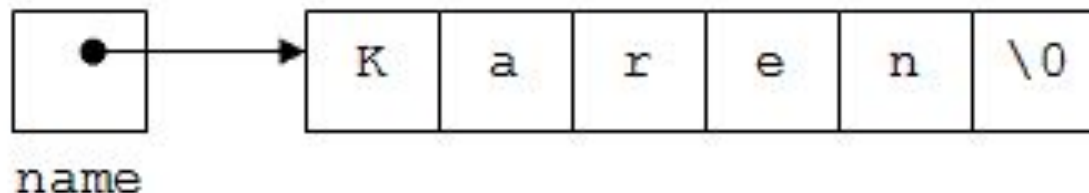
Apresentação 09

Roteiro

- Strings
- Matrizes
- Atividade Prática

O termo “String” serve para designar um conjunto de caracteres. Na prática, uma string serve para representar um informação texto.

Algumas linguagens de programação não possuem o tipo nativo String. No C o que existe são os conjuntos (arrays) de caracteres.



Declaração de uma String:

```
char nome[30];  
char nome[];
```

Inicialização de uma String:

```
char nome[30] = "Alguem";  
char nome[30] = { 'A', 'l', 'g', 'u', 'e', 'm' };
```


Leitura de uma String:

Podemos ler uma string com os comandos scanf ou ainda com gets.

`scanf("%s",&nome);`

A função scanf realiza a leitura até encontrar um espaço, depois encerra a leitura e coloca o caracter terminador \0.

A variável que vai armazenar a string não necessita ser precedida por &.

`gets(nome);`

Esta função armazena tudo que foi digitado, inclusive os espaços, até que a tecla ENTER seja pressionada

Exemplo:

Programa C que lê nome de uma cidade e imprime individualmente cada letra:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  //biblioteca de funções para strings
4  #include <string.h>
5
6  int main(int argc, char *argv[]) {
7
8      char cidade[30];
9      int cont, tamanho;
10
11     printf("Informe a cidade de nascimento:");
12     gets(cidade);
13     //strlen -> retorna o tamanho de uma string
14     tamanho = strlen(cidade);
15     for(cont=0; cont<tamanho; cont++)
16     {
17         printf("\n %c", cidade[cont]);
18     }
19     return 0;
20 }
```

A biblioteca string.h reúne diversas funções úteis para a manipulação de strings:

strcpy

Realiza a cópia do conteúdo de uma variável a outra.

Sintaxe:

```
strcpy(string_destino, string_origem);
```

strlen

Retorna o tamanho de uma string.

Sintaxe:

```
variável_tipo_inteiro = strlen(string_origem);
```

A biblioteca string.h reúne diversas funções úteis para a manipulação de strings:

strcat

Concatena (une) o conteúdo de 2 strings (sempre na destino).

Sintaxe:

```
strcat(string_destino, string_origem);
```

strcmp

Compara o conteúdo de 2 strings.

Sintaxe:

```
variável_tipo_inteiro = strcmp(string1, string2);
```

Retorna 0 -> se forem iguais



BIBLIOTECA STRING.H

Exemplo:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  //biblioteca de funções para strings
4  #include <string.h>
5
6  int main(int argc, char *argv[]) {
7
8      char palavra1[30], palavra2[30];
9      int cont, tamanho,iguais;
10
11     printf("Informe a palavra1: ");
12     gets(palavra1);
13     printf("Informe a palavra2: ");
14     gets(palavra2);
15     //Guarda o tamanho da palavra1
16     tamanho = strlen(palavra1);
17     printf("Tamanho da palavra1 e: %d",tamanho);
18     //Compara a a palavra1 com a palavra2
19     iguais = strcmp(palavra1,palavra2);
20     if(iguais==0)
21     {
22         printf("\nAs palavras são iguais");
23     }
24     else
25     {
26         printf("\nAs palavras são diferentes");
27     }
28     //palavra1 passa a ser "Fadergs"
29     strcpy(palavra1, "Fadergs");
30     printf("\nA palavra1 agora e: %s",palavra1);
31     //Acrescenta a palavra2 dentro da palavra1
32     strcat(palavra1, palavra2);
33     printf("\nA palavra2 agora e: %s",palavra2);
34     return 0;
35 }
```

A biblioteca `string.h` reúne diversas funções úteis para a manipulação de strings:

toupper

Transforma um caracter em maiúsculo.

Sintaxe:

```
toupper(caracter) ;
```

tolower

Transforma um caracter em minúsculo.

Sintaxe:

```
tolower(caracter) ;
```

OUTRAS FUNÇÕES

```

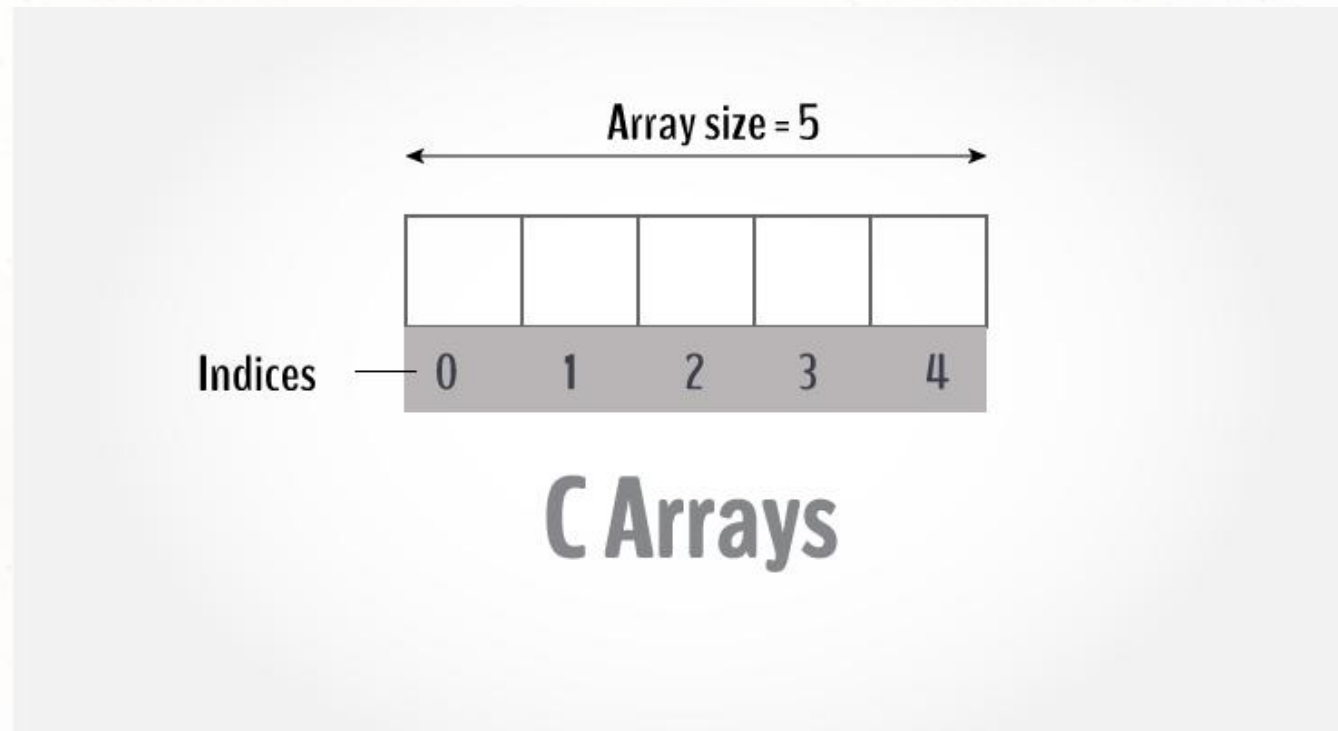
1  #include <stdio.h>
2  #include <strings.h>
3  #include <ctype.h>
4
5  int main() {
6      char cidade[20]="porto alegre";
7      int x;
8
9      for(x=0;x<strlen(cidade);x++)
10     {
11         printf("%c",toupper(cidade[x]));
12     }
13 }

```




Variáveis Compostas Homogêneas Multidimensionais

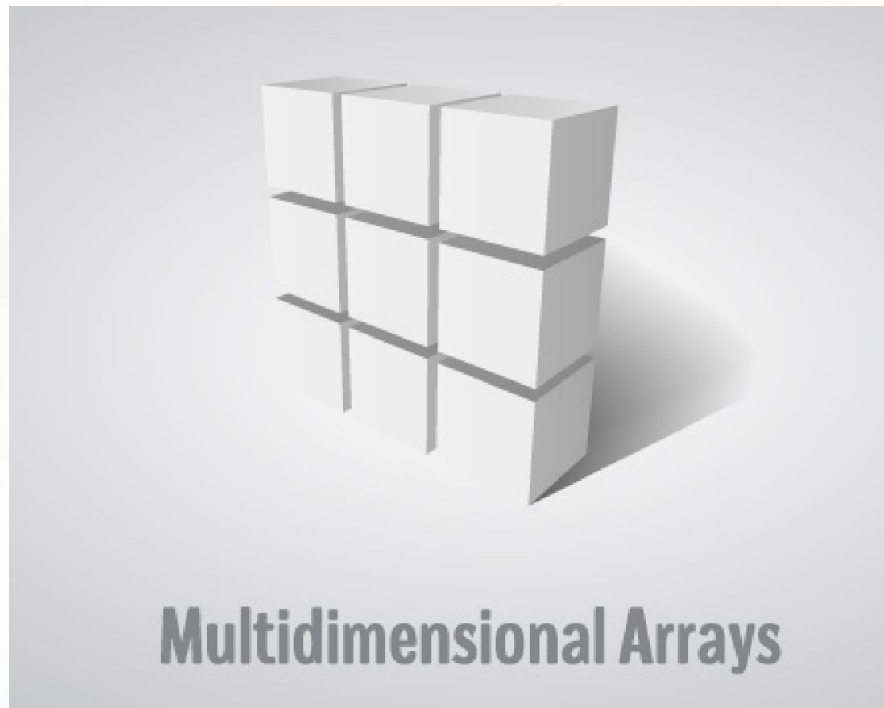
Array Unidimensional:



Variáveis Compostas Homogêneas Multidimensionais

Vetores ou variáveis compostas homogêneas unidimensionais são estruturas bastante utilizadas nos mais diversos tipos de aplicações. Porém, em alguns casos é necessário trabalharmos com mais de uma dimensão.

Variáveis Compostas Homogêneas Multidimensionais são chamadas de Matrizes.





Variáveis Compostas Homogêneas Multidimensionais

Declaração de uma Matriz Bidimensional:

```
char velha[3][3];  
char velha[][];
```

Inicialização de uma Matriz Bidimensional:

```
char velha[3][3] = {{ 'X' , '0' , 'X' } ,  
                    { '0' , 'X' , '0' }  
                    { 'X' , '0' , 'X' } };
```

Variáveis Compostas Homogêneas Multidimensionais

Exemplo:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  //biblioteca de funções para strings
4  #include <string.h>
5
6  int main(int argc, char *argv[]) {
7
8      int lin,col;
9      char velha[3][3] = {{ ' ', ' ', ' ', ' ' },
10                          { ' ', ' ', ' ', ' ' },
11                          { ' ', ' ', ' ', ' ' }};
12
13      for(lin=0;lin<3;lin++)
14      {
15          for(col=0;col<3;col++)
16          {
17              printf("\nInforme o 'X' ou 'O' na posicao %d,%d: ",lin,col);
18              scanf(" %c",&velha[lin][col]); //espaco a frente de %c
19          }
20      }
21
22      for(lin=0;lin<3;lin++)
23      {
24          for(col=0;col<3;col++)
25          {
26              printf("%c ",velha[lin][col]);
27          }
28          printf("\n");
29      }
30      return 0;
31  }

```



Variáveis Compostas Homogêneas Multidimensionais

Tridimensionais:



tipo cubo[2][2][2];



tipo cubo[3][3][3];

1. *Crie um programa em C que leia uma frase e substitua letras 'a' por letras 'o'.*
2. *Crie um programa em C que leia uma frase e inverta as letras dessa frase.*
3. *Crie um programa em C que leia uma frase e logo leia uma letra. A letra informada deverá ser eliminada em todas as posições que apareça nessa frase.*
*Ex: **Digite uma frase:** analise e desenvolvimento de sistemas*
***Digite uma letra:** e*
***Resultado:** analis dsnvolvimnto d sistmas*

4. *Crie um código C que receba uma frase e um caractere, e retorne o número de vezes que esse caractere aparece na string.*
5. *Crie um código C que receba uma frase e mostre os caracteres de uma string são repetidos*
6. *Crie um código C que receba uma frase e coloque a inicial de cada palavra em maiúsculo.*
7. *Crie um código C repita os caracteres de uma frase informada.
Ex.: <usuário digita> Fadergs
<impressão na tela> FFaaddeerrss*
8. *Escreva um programa que lê uma string e retorna verdadeiro se a string de entrada é uma palíndrome.*
9. *Codifique um programa que recebe uma string e substitui cada letra pela letra seguinte circularmente (A é substituído por B, ... e Z é substituído por A). Escreva funções para codificar e decodificar strings.*

10. *Escreva um programa que funciona como o jogo da forca. O usuário tenta acertar uma palavra (contida numa variável) usando 5 tentativas. A palavra é mostrada ao usuário com as letras marcadas com o símbolo '*'. O usuário deve digitar uma letra (sem acentos, til, trema) e caso a letra esteja correta, o símbolo '*' é substituído pela letra. Se o usuário não montar a palavra até a quinta tentativa, o jogo acaba em derrota.*

- 11.** *Crie um programa em C que leia uma matriz 3x3 de inteiros e depois multiplique todos os elementos dessa matriz por 10. Logo, imprima a matriz resultante.*
- 12.** *Crie um programa em C que leia uma matriz 3x3 de inteiros e depois divida todos os números pares por 2. Logo, imprima a matriz resultante.*
- 13.** *Crie um programa em C que leia uma matriz 3x3x3 de inteiros e depois multiplique todos os números ímpares por 5. Logo, imprima a matriz resultante.*



ecosistema
ănimă