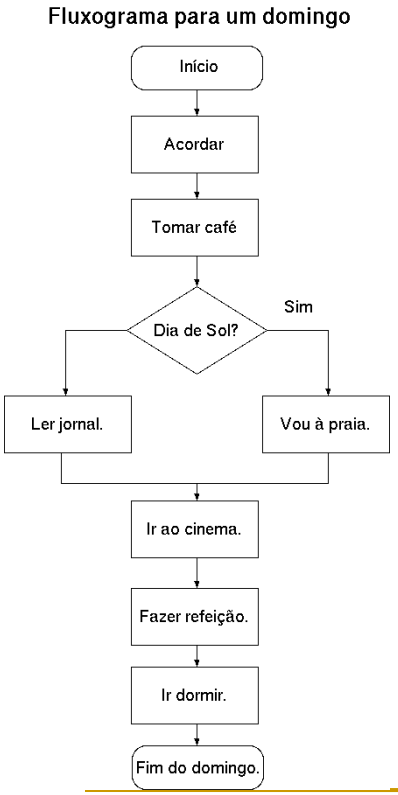
UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto



Ciência da Computação

Prof. Dr. Leandro Alves Neves

Aula 08

Algoritmos e Programação

Sumário

- String
  - Manipulação de palavras em array (vetores)
  - Manipulação de strings (string.h)
  - Busca de palavras em texto (string matching)

Definição

- String
  - Sequência de caracteres adjacentes na memória.
  - Strings: arrays (vetores) do tipo char.

- Ex:
  - char str[6];

Definição

- String: Característica
  - O elemento seguinte a última letra da palavra/frase armazenada é um caractere '\0'.
  - O caractere ‘\0’ indica o fim da sequência.

- Ex: char str[6] = “Oi”;

O	i	\0

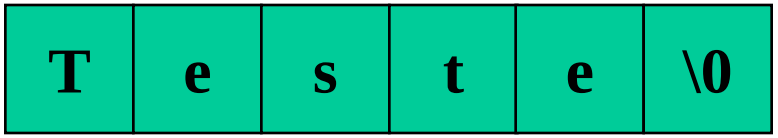
Definição

- Importante
  - Na declaração de uma string, é necessário considerar o caractere ‘\0’.
  - Isso significa que a string *str* permite o armazenamento de no máximo 5 caracteres.
- Ex:
  - char str[6] = “Teste”;

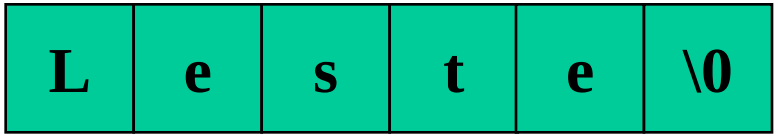
T	e	s	t	e	\0

Definição

- Por ser tratada como array, cada caractere pode ser acessado individualmente (via indexação)
  - Ex: char str[6] =

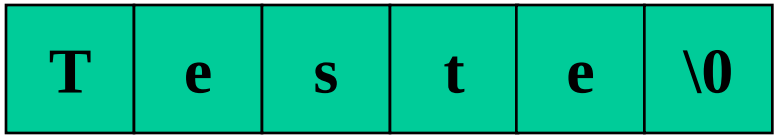


  - str[0] = ‘L’;

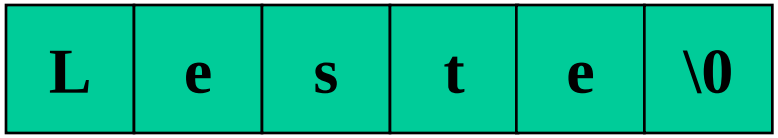


Definição

- IMPORTANTE:
  - Na inicialização de palavras, usa-se “aspas duplas”.
    - Ex: char str[6] = “Teste”;



  - Na atribuição de caracteres, usa-se ‘aspas simples’
    - str[0] = ‘L’;



Manipulando strings

- Strings são arrays. Portanto, não se pode atribuir uma string para outra!
  -

```c
#include <stdio.h>

int main()
{
  char str1[]="Algoritmos e Técnicas";
  //inclui automaticamente o '\0'
  char str2[20];

  str1=str2; // ERRADO!
  puts(str1);
  return 0;
}
```

- O correto é copiar a string elemento por elemento.

Manipulando strings

- Entrada de Dados:
  - É possível usar scanf para entrada de uma string
    - Exemplo:

```c
#include <stdio.h>

int main()
{
    char str1[21];
    puts("\nDigite uma palavra: ");
    scanf("%s",str1);
    puts(str1);
    return 0;
}
```

  - Nesse caso, o nome do vetor é o endereço para a primeira posição da estrutura array

Manipulando strings

- Entrada de Dados:
  - O uso de scanf limita entrada de espaço
  - Solução: fgets
    - Exemplo:

```c
#include <stdio.h>
#define size 21
int main()
{
  char str1[size];
  puts("\nDigite uma palavra: ");

  fgets(str1, size, stdin);
  puts(str1);
  return 0;
}
```

  - Nesse caso, espaço é considerado *string*, impondo uma tamanho limitado de entrada.

Manipulando strings

- A biblioteca padrão C possui funções especialmente desenvolvidas para esse tipo de tarefa

- #include <string.h>

Manipulando strings - Tamanho

- strlen(str):
  - Retorna o tamanho (unsigned long) da string str.

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

int main()
{
  char str1[]="Algoritmos e ", str2[]="Técnicas de Prog.";

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  printf("\nTamanho de str1: %lu",strlen(str1));
  printf("\nTamanho de str2: %lu",strlen(str2));

  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

    - Manipulação de String -

str 1: Algoritmos e
str 2: Técnicas de Prog.
Tamanho de str1: 13
Tamanho de str2: 18
```

Manipulando strings - Copiar

- strcpy(dest, fonte):
  - Copia uma string contida na variável *fonte* para *dest*.

```c
#include <stdio.h>
#include <string.h>
#define size 41
int main()
{
  char str1[]="Algoritmos e ", str2[]="Técnicas de Prog.";
  char str3[size];

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  printf("\nTamanho de str1: %lu",strlen(str1));
  printf("\nTamanho de str2: %lu",strlen(str2));
  strcpy(str3,str1);
  printf("\n%s",str3);
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

     - Manipulação de String -

str 1: Algoritmos e
str 2: Técnicas de Prog.
Tamanho de str1: 13
Tamanho de str2: 18
Algoritmos e
```

Manipulando strings - Concatenar

- strcat(dest, fonte):
  - Concatena duas strings. Nesse caso, a string contida em *fonte* permanecerá inalterada e será anexada ao final da string *dest*.

```c
#include <stdio.h>
#include <string.h>
#define size 41
int main()
{
  char str1[]="Algoritmos e ", str2[]="Técnicas de Prog.";
  char str3[size];

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  printf("\nTamanho de str1: %lu",strlen(str1));
  printf("\nTamanho de str2: %lu",strlen(str2));
  strcpy(str3,str1);
  printf("\n%s",str3);
  strcat(str3,str2);
  printf("\n%s",str3);
  printf("\nTamanho de str3: %lu",strlen(str3));
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

     - Manipulação de String -

str 1: Algoritmos e
str 2: Técnicas de Prog.
Tamanho de str1: 13
Tamanho de str2: 19
Algoritmos e
Algoritmos e Técnicas de Prog.
Tamanho de str3: 32
```

Manipulando strings - Comparar

- strcmp(str1, str2):
  - Compara duas strings. O retorno da função pode ser:
    - 0: para strings iguais;
    - <: string 1 é menor que a string 2;
    - >: string 1 é maior que a string 2.

Manipulando strings - Comparar

- strcmp(str1, str2):
  - Compara duas strings. O retorno da função pode ser:

```c
#include <stdio.h>
#include <string.h>
#define size 41
int main()
{
  char str1[]="Algoritmos", str2[]="Algoritmos";

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  if (strcmp(str1,str2)==0)
    puts("\nstr1 e str2 são iguais!");
  else if (strcmp(str1,str2)<0)
    puts("\nstr1 é menor que str2!");
  else
    puts("\nstr1 é maior que str2!");
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

     - Manipulação de String -

str 1: Algoritmos
str 2: Algoritmos
str1 e str2 são iguais!
```

Manipulando strings - Comparar

- strcmp(str1, str2):
  - Compara duas strings. O retorno da função pode ser:

```c
#include <stdio.h>
#include <string.h>
#define size 41
int main()
{
  char str1[]="Algoritmos", str2[]="Algoritmos e Téc.";

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  if (strcmp(str1,str2)==0)
    puts("\nstr1 e str2 são iguais!");
  else if (strcmp(str1,str2)<0)
    puts("\nstr1 é menor que str2!");
  else
    puts("\nstr1 é maior que str2!");
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

     - Manipulação de String -

str 1: Algoritmos
str 2: Algoritmos e Téc.
str1 é menor que str2!
```

Manipulando strings - Comparar

- strcmp(str1, str2):
  - Compara duas strings. O retorno da função pode ser:

```c
#include <stdio.h>
#include <string.h>
#define size 41
int main()
{
  char str1[]="Algoritmos e Téc.", str2[]="Algoritmos";

  puts("\n\t - Manipulação de String -");

  printf("\nstr 1: %s", str1);
  printf("\nstr 2: %s", str2);

  if (strcmp(str1,str2)==0)
    puts("\nstr1 e str2 são iguais!");
  else if (strcmp(str1,str2)<0)
    puts("\nstr1 é menor que str2!");
  else
    puts("\nstr1 é maior que str2!");
  return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main

      - Manipulação de String -

str 1: Algoritmos e Téc.
str 2: Algoritmos
str1 é maior que str2!
```

Bibliografia Complementar

- SCHILDT, H. C Completo e Total, 3ª ed., Pearson 1996. 852p.
  - Páginas 294,295, 347, 349 e 351.