Algoritmos e Lógica de Programação

80 horas // 4 h/semana

Strings ou cadeias de Caracteres

Aula 14

Prof. Piva

Para começar...

- *Strings* são cadeias de caracteres que armazenam dados textuais e, portanto, podem armazenar informações para as mais diversas finalidades.
- O conteúdo de uma *string* pode representar um fato em si, ou uma informação. Por exemplo, se uma *string* armazena um valor igual a 120, isso é um dado, que somente será entendido conhecendo seu contexto.
- Agora, se uma String armazena a frase: "Neste mês vendemos 120 motores, e isso foi muito bom, pois significou um aumento de 30% nas vendas desse produto, em relação ao mesmo período do ano passado.". Temos, então, uma informação armazenada na **string**.

Para começar...

- Podemos, a partir desses exemplos simples, imaginar a importância desse tipo de variável *Strings*, na construção de algoritmos ou programas de computadores.
- A partir do estabelecimento de relações entre dados, contidos em *strings*, podemos gerar informação, que por sua vez poderá, a partir de outros relacionamentos, gerar conhecimento.
- A maioria dos mecanismos de busca, que conhecemos na Internet, funcionam manipulando extensas cadeias de caracteres, contidas em milhares de bases de dados nessa rede.
- Essas cadeias de caracteres, contidas em textos curtos ou longos, são denominadas *strings*.

Strings

- Vamos aprender a manipular variáveis do tipo *string*, atribuindo valores a mesma, e recuperando seu conteúdo, usando linguagem algorítmica ou de programação de computador.
- Dependendo da linguagem de programação, *string* pode ser um tipo de dado primitivo, uma classe, ou mesmo um tipo criado pelo programador.
- A *string* referencia como "cadeia de caracteres" e, portanto, poder ser visualizada como uma lista linear ou vetor.
- Cada elemento da string é um caractere, e o agrupamento deles irá representar um dado ou uma informação.

Strings

- Em *strings*, os caracteres são armazenados da esquerda para a direita.

  Exemplo de uma *string* (uma frase) contendo 45 posições:

Exemplo...

- Considerando a *string* do exemplo anterior, vamos construir trechos de algoritmos em VisuALG para manipular essa *string*.

Exemplo... VisuAlg

- Definir uma variável do tipo de dado *string* e atribuir à ela, a frase do exemplo anterior. Definir outra variável do tipo *string* e atribuir à ela, a frase: "Conforme Salvetti (2000, p. 3), ". Finalmente, exibir o conteúdo dessas duas strings, formando a frase: "**Conforme Salvetti (2000, p. 3), PARA CONSTRUIR ALGORITMOS BASTA LÁPIS E PAPEL**". Atenção! - o sinal "+" é usado para concatenar duas, ou mais, *strings*.

```
algoritmo "Frase"
Var
  Str_Frase1, Str_Frase2 : Caractere
Inicio
  LimpaTela
  Str_Frase1 := "PARA CONSTRUIR ALGORITMOS BASTA
  LÁPIS E PAPEL"
  Str_Frase2 := "Conforme Salvetti (2000, p. 3), "
  Escreval(Str_Frase2 + Str_Frase1)
fimalgoritmo
```

Exemplo... VisuAlg

- Note que faltou terminar a frase com ponto final. Neste caso, podemos acrescentar o ponto no final da Str_Frase1, ou concatena-lo com as *strings*, conforme mostrado no programa abaixo.

```
algoritmo "Frase"
Var
  Str_Frase1, Str_Frase2 : Caractere
Inicio
  LimpaTela
  Str_Frase1 := "PARA CONSTRUIR ALGORITMOS BASTA
  LÁPIS E PAPEL"
  Str_Frase2 := "Conforme Salvetti (2000, p. 3), "
  Escreval(Str_Frase2 + Str_Frase1 + ". ")
fimalgoritmo
```

Strings – algumas funções em VisuALG

Sub-rotina - VisuAlg
Funcionalidade
asc(cadeia:caractere):inteiro
Função que retorna um inteiro com o código ASCII, em decimal, do primeiro caractere contido na cadeia de caractere.
compr(cadeia:caractere):inteiro
Função que retorna um inteiro contendo o comprimento (quantidade de caracteres) da cadeia.
maiusc(cadeia:caractere):caractere
Função que retorna cada caractere da cadeia convertido para maiúsculos.

Strings – algumas funções em VisuALG

minusc(cadeia:caractere):caractere
Função que retorna cada caractere da cadeia convertido para minúsculos.
pos(subcadeia,cadeia:caractere):inteiro
Função que retorna um inteiro que indica a posição em que a subcadeia de caractere se encontra dentro da cadeia, ou zero se subcadeia não estiver contida na cadeia.
numpcarac(numero:inteiro ou real):caractere
Função que converte um número inteiro ou real para uma cadeia de caractere.

Strings – algumas funções em VisuALG

caracpnum(cadeia: caractere):inteiro ou real
Função que converte uma cadeia de caractere em um número inteiro ou real.
copia(cadeia:caractere;pos,qtdcarac: inteiro):caractere
Função que retorna um valor do tipo caractere contendo uma cópia parcial da cadeia, a partir da posição indicada pela variável pos, com a quantidade de caracteres indicada pela variável qtdcarac. Os caracteres da cadeia são numerados da esquerda para a direita, começando de 1.

Variável String

- Matriz unidimensional (vetor) do tipo char terminada pelo caractere null '\0'
- cada caractere de um string pode ser acessado individualmente
- vetor de tamanho n → string de tamanho (n-1)

Ex:

```
char  string[10]  =  "exemplo" ;
char  string[10] = { "exemplo" };
char  string[10]  = { 'e', 'x', 'e', 'm', 'p', 'l', 'o', '\0' };

 printf ( "%s", string );
 printf ( "%c", string [ 0 ] );
```

Lendo Strings

- scanf : lê o string até que um branco seja encontrado

Ex:
```
main ( )
{
        char  nome[40];

        printf ( “Digite seu nome: “ );
        scanf ( “%s”, &nome[ 0 ] );
        //scanf ( “%s”, nome );
        printf ( “Bom dia %c”, nome[0] );
}
```

Saída:
Digite seu nome: **Jose Maria**
Bom dia **Jose**

Lendo Strings

- Gets
  lê caracteres até encontrar ‘\n’
  substitui ‘\n’ por ‘\0’

Ex:
```
main ( )
{
        char  nome[40];

        printf ( “Digite seu nome: “ );
        gets( &nome[ 0 ] ); // ou  gets(nome);
        printf ( “Bom dia %s”, nome );
}
```

Saída:
Digite seu nome: **Jose Maria**
Bom dia **Jose Maria**

Imprimindo Strings

- printf
- puts

Ex:
```
main ( )
{
        char  nome[40];

        printf ( “Digite seu nome: “ );
        gets ( &nome[ 0 ] );
        puts ( “Bom dia ” );
        puts ( nome );
}
```

Saída:
Digite seu nome: **Jose Maria**
Bom dia
Jose Maria

Funções de manipulação de strings

- Strlen
  retorna o tamanho do string - não conta ‘\0’

Ex:
```
main ( )
{
        char  nome[40];

        printf ( “Digite seu nome: “ );
        gets ( &nome[ 0 ] );
        printf (“Tamanho = %d”, strlen(&nome[ 0 ]) );
}
```

Saída:
Digite seu nome: **Jose Maria**
Tamanho = 10

Funções de manipulação de strings

- strcat (str1, str2)
  concatena str2 ao final de str1

Ex:
```
main ( )
{
        char    nome[40]  = “Jose”,
        char sobrenome[30] =  “Maria”;

        strcat(nome, sobrenome);
        puts (sobrenome);
        puts (nome);
}
```

Saída:
Maria
JoseMaria
Cuidado:
dado **str1 + str2**
tem que caber em
str1

Funções de manipulação de strings

- strcmp (str1, str2)
- compara dois strings retornando:
  - negativo se str1 < str2
  - 0 se str1 = str2
  - positivo se str1 > str2
- a comparação é feita por ordem alfabética

```
#include <stdio.h>
main ( )
  {
      char  nome[40]  = “Jose”;
      char  sobrenome[30] =  “Maria”;

      if  ( strcmp ( nome, sobrenome ) !=0)
         puts ( “os strings são diferentes” );
      else
         puts ( “os strings são identicos” );
  }
```

Strings – Algumas funções em C

Sub-rotina - C Funcionalidade
int strlen(cadeia)
Função que retorna o número de caracteres da armazenado na *cadeia*, não considerando o caractere NULL (/0).
int strcmp(cadeia1,cadeia2)
Função que retorna um valor 0 (zero) se as duas cadeias são iguais.
char strupr(cadeia);
Função que retorna cada caractere da cadeia convertido para maiúsculos.
char strlwr(cadeia);
Função que retorna cada caractere da cadeia convertido para minúsculos.

Strings – Algumas funções em C

char strcat(cadeia1,cadeia2);
Função que retorna uma cadeia resultante da união entre duas cadeias passadas como parâmetro na função.
char strcpy(cadeia1, cadeia2);
Função que copia o conteúdo da cadeia2 para dentro da cadeia1. A cadeia 2 pode ser uma constante.
char strncpy(cadeia1,cadeia2,int qtdcarac);
Função que armazena na cadeia1 os primeiros caracteres da cadeia2, cuja quantidade está indicada em qtdcarac. Esse segundo parâmetro (qtdcarac) pode ser uma constante numérica inteira sem sinal. Atenção! O caractere NULL não é armazenado, devendo isto ser feito pelo programa.

Strings – Algumas funções em C

Sub-rotina - C Funcionalidade
char strstr(cadeia1,cadeia2);
Função que verifica se a cadeia2 é subcadeia da cadeia1. Retorna um ponteiro para a primeira posição a partir da qual a cadeia2 ocorre na cadeia1. Retorna NULL se cadeia2 não está contida na cadeia1.
int atoi(cadeia); long atol(cadeia); double atof(cadeia);
Funções para conversão de números passados como *strings*. As funções acima convertem uma cadeia de dígitos, respectivamente, para os tipos *int*, *long* ou *float*. Essas funções retornam o número (no formato respectivo) correspondente à primeira posição (da esquerda para direita) da cadeia que pode ser convertida. Retorna 0(zero) se esse primeiro caractere da cadeia não for um dígito, ou um dos caracteres + e - (se o primeiro caractere for + ou -, para que haja alguma conversão o segundo deve ser um dígito). Esse procedimento se repete com todos os elementos da cadeia. Atenção! Essas funções estão na biblioteca stdlib.h.

EXERCÍCIOS

Strings...

EXERCÍCIO 1

Elabore um algoritmo para ler/receber, separadamente, o primeiro nome, o nome do meio e o sobrenome de uma pessoa, e mostre o nome completo, correspondente.

EXERCÍCIO 2

Faça um algoritmo que solicite uma data no formato de uma string – dd/mm/aaaa. Mostre essa data no formato AAAAMMDD

EXERCÍCIO 3

Faça um algoritmo para ler nove caracteres numéricos em uma string. Mostre o conteúdo dessa string colando pontos e virgula, respectivamente nas posições inteiras e decimais.
Exemplo:
Digitado> 987654321
Mostrado> 9.876.543,21

EXERCÍCIO 4

Elabore um algoritmo para determinar quantas vogais existem dentro de uma determinada frase (que deve ser recebida do usuário).

EXERCÍCIO 5

Faça um algoritmo para determinar quantas palavras existem em uma determinada frase Obs: tanto a palavra, quanto a frase, devem ser informadas pelo usuário.

EXERCÍCIO 6

Faça um algoritmo para determinar se um determinado vetor, digitado pelo usuário, é um palíndromo.

Palíndromo: lido da direita para a esquerda, ou vice versa, representam a mesma coisa.
Ex: AMA