Sistemas para Internet
UFSM

Strings

Professor Leandro O. Freitas
leandro@politecnico.ufsm.br

Retomando...

Sistemas para Internet
UFSM

- Vetores

```
void exemplo_Vetor() {
    int i;
    int v[3] = {10, 20, 30};
    for (i=0; i<3; i++){
        printf("%d ", v[i]);
    }
}
```

- Matrizes

```
void exemplo_Matriz() {
    int i, j;
    int m[2][2] = {2,4,6,8};
    for (i=0; i<2; i++)
    {
        for (j=0; j<2; j++)
        {
         printf("%d ", m[i][j]);
        }
    }
}
```

Strings

Sistemas para Internet
UFSM

- Conjunto de caracteres que aparecem entre aspas;
- Em C não existe um tipo de variável específico para string;
- Como armazenar uma palavra em uma variável?
- Cria-se um vetor do tipo ***char*** onde cada posição corresponde a uma letra do texto.
  - Exemplo:

```
char a[5] = {'a', 'u', 'l', 'a'};
```

Strings

Sistemas para Internet
UFSM

- Mas como prever o final do texto, ou seja, o tamanho do vetor?

- Utilizamos um caractere nulo (‘ \0 ’) para simbolizar o final da string.
  - Ao digitar um texto e pressionar ENTER, o caractere nulo é inserido automaticamente após a última letra.

- Então: string é um conjunto de caracteres alfanuméricos armazenados em um vetor do tipo **char**, onde o caractere nulo (‘ \0 ’) representa o fim da cadeia.

- ATENÇÃO: O caractere nulo também ocupa uma posição no vetor!
  - Devemos declarar o vetor com uma posição a mais para armazenar o caractere nulo.

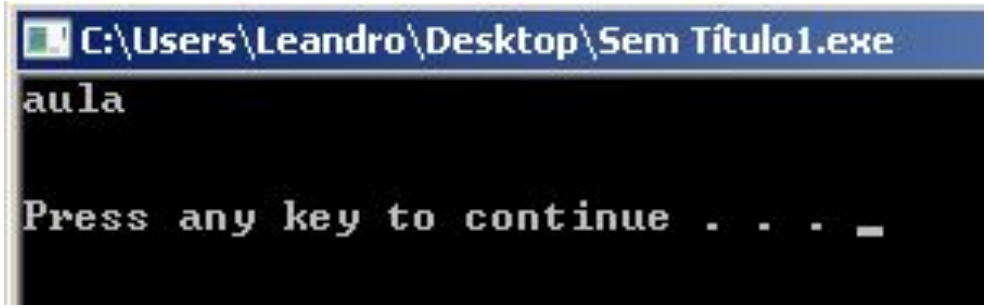
Exemplo

Sistemas para Internet
UFSM

```
char a[20];

a[0] = 'a';
a[1] = 'u';
a[2] = 'l';
a[3] = 'a';
a[4] = '\0';
a[5] = 's';
```

```
char a[20] = {'a', 'u', 'l', 'a', '\0', 's'};
```

Considerando '\0' o final de uma string, o que será impresso ao executar este trecho de código?



Operações inválidas

Sistemas para Internet
UFSM

- String não é um tipo primitivo do C, logo existem operações que não podem ser realizadas:

```
char a[20];
char b[20] = {"aula de algoritmos"}; // correto!

a = b;  // Errado! Não é possível fazer atribuição de strings dessa forma!

if (a == b) // Errado! Não é possível fazer comparação de strings dessa forma!

a = "casa"; // Errado! Não é possível fazer atribuição de strings dessa forma!
```

Funções de manipulação de Strings

- Biblioteca: *string.h*
- Para referenciar: %s

- Funções:
  - gets()
  - puts()
  - strcpy()
  - strcat()
  - strcmp()
  - strlen()
  - strupr()
  - strlwr()

Problemas da função scanf()

Sistemas para Internet
UFSM

- A função scanf() lê um valor até o primeiro espaço informado e adiciona o caractere nulo (‘\0’).

```
#include<string.h>

void exemploStrings()
{
    char a[20];

    printf("digite seu nome: ");
    scanf("%s",&a);

    printf("Seu nome eh: %s",a);

    printf("\n\n");

}
```

```
digite seu nome: pedro silva
Seu nome eh: pedro

Press any key to continue . . .
```

- Não testa o tamanho do vetor.

Função gets(): leitura de dados

Sistemas para Internet
UFSM

- Tem o mesmo objetivo da função scanf(), com a vantagem de:
  - Armazenar textos até que a tecla **Enter** seja pressionada, ou seja, armazenar textos com espaços em branco;

- Exemplo:

```
char Endereco[50];
gets(Endereco);
```

- Problema: A função gets() permite que sejam armazenados caracteres além da capacidade do vetor, o que pode provocar erros no programa. **Pesquise sobre a função fgets!**

- *"warning: the 'gets' function is dangerous and should not be used."*

Função puts(): saída de dados

Sistemas para Internet
UFSM

- Tem o mesmo objetivo da função **printf**;
- Sintaxe mais simples: puts (nome_da_variável);
- Não permite a inserção de textos e variáveis (concatenação).

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

void exemploStrings()
{
    char a[10];

    printf("digite seu nome: ");
    gets(a);

    puts(a);

    printf("\n\n");

}
```

```
digite seu nome: leandro
leandro


Press any key to continue . . .
```

Exercício 1

Sistemas para Internet
UFSM

- Desenvolva um algoritmo com uma função para ler seu nome e armazená-lo na variável **nome** e seu sobrenome armazenando-o na variável **sobrenome**. Depois mostrar o valor das duas variáveis para o usuário.

Exercício 1

Sistemas para Internet
UFSM

- Solução:

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

void exemplo1(){
    char nome[20], snome[20];
    printf("digite seu nome: ");
    gets(nome);
    printf("digite seu sobrenome: ");
    gets(snome);
    puts(nome);
    puts(snome);

}

int main()
{
    exemplo1();

    return 0;
}
```

Função strcpy()

Sistemas para Internet
UFSM

- Copia (atribui) o conteúdo de uma string para outra;
- Semelhante ao operador de atribuição (=), para valores numéricos;
- Sintaxe:

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

- Onde:

  string_destino é a variável que receberá o conteúdo de string_origem;

  string_origem é a variável que possui o conteúdo que será copiado.

- A **string_destino** deve ter tamanho suficiente para receber todo o conteúdo da **string_origem**.

Função strcpy(): exemplos

Sistemas para Internet
UFSM

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

void exemploStrings()
{
    char a[20] = {"aula de algoritmos"};
    char b[20];

    strcpy(b, a);

    puts(b);

    printf("\n\n");

}
```

```
aula de algoritmos

Press any key to continue . . .
```

Função strcat()

Sistemas para Internet
UFSM

- Concatena (une) o conteúdo de duas strings.
- Sintaxe:

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

- Onde:

  string_destino recebe conteúdo de **string_origem** anexando-o ao final de seu próprio conteúdo;

  string_origem possui o conteúdo que será concatenado na **string_destino**.

- A **string_destino** deve ter tamanho suficiente para receber todo o conteúdo da **string_origem**.

Função strcat(): exemplos

Sistemas para Internet
UFSM

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

void exemploStrings()
{
    char a[20] = {"de algoritmos"};
    char b[20] = {"aula "};

    strcat(b, a);

    puts(b);

    printf("\n\n");

}
```

```
aula de algoritmos

Press any key to continue . . .
```

Exercício 2

Sistemas para Internet
UFSM

- Faça as seguintes alterações no exercício 1:
  - Crie um terceiro vetor chamado **nome_completo** e copie o conteúdo da variável **nome** para ele.
  - Concatene o conteúdo da variável **sobrenome** para o final da variável **nome_completo**.
  - Imprima o conteúdo da variável **nome_completo** para o usuário.

Qual o problema que surge e como solucioná-lo??

Exercício 2

Sistemas para Internet
UFSM

- Solução:

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

void exemplo1(){
    char nome[20], snome[20];
    printf("digite seu nome: ");
    gets(nome);
    printf("digite seu sobrenome: ");
    gets(snome);
    char nome_completo[40];
    strcpy(nome_completo, nome);
    strcat(nome_completo, " ");
    strcat(nome_completo, snome);
    puts(nome_completo);

}

int main()
{
    exemplo1();
    return 0;
}
```

Função strcmp()

Sistemas para Internet
UFSM

- Compara o conteúdo de duas strings;
- Caso os conteúdos sejam exatamente iguais, a função retorna zero. Caso contrário retorna um valor diferente de zero;
- Esta função diferencia letras maiúsculas e minúsculas, logo:

Algoritmos != algoritmos

Função strcmp(): exemplos

Sistemas para Internet
UFSM

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

void exemploStrings()
{
    char a[20];
    char b[20];
    int retorno;

    gets(a);
    gets(b);

    retorno = strcmp(b, a);

    if (retorno == 0)
        printf("As strings sao iguais.");

    else
        printf ("As strings sao diferentes");
}
```

Função strlen()

Sistemas para Internet
UFSM

- Retorna o comprimento da string:
  - Quantidade de caracteres que ela possui.
- O caractere terminador nulo ('\0') não é contabilizado.

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

void exemploStrings()
{
    char a[20];
    int tamanho;

    gets(a);

    tamanho = strlen(a);

    printf("%d",tamanho);
}
```

Exercício 3

Sistemas para Internet
UFSM

- Crie um algoritmo com uma função que contenha dois vetores do tipo char, chamados **palavra1** e **palavra2**. Utilizando a função gets, atribua valores (texto) para elas e depois faça o seguinte:
  - Descubra o tamanho de cada uma delas.
  - Se e, somente, se tiverem o mesmo tamanho, verifique se as duas palavras são iguais e informe ao usuário sobre isso.

Exercício 3

Sistemas para Internet
UFSM

- Solução:

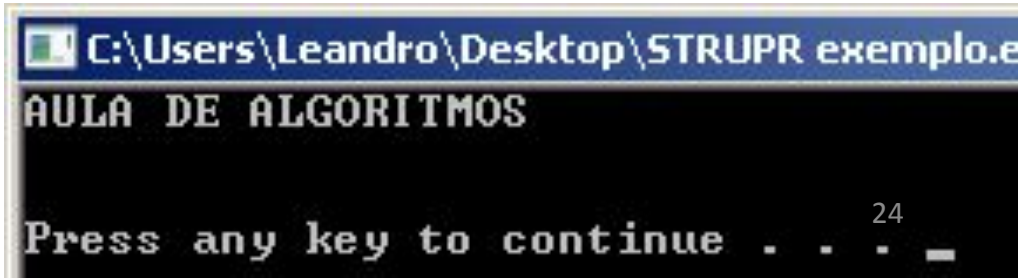
```c
#include <stdio.h>
#include <string.h>

void exemplo1(){
    char palavra1[20], palavra2[20];
    printf("digite a palavra 1: ");
    gets(palavra1);
    printf("digite a palavra 2: ");
    gets(palavra2);
    int tam_p1, tam_p2;
    tam_p1 = strlen(palavra1);
    tam_p2 = strlen(palavra2);
    if(tam_p1 == tam_p2){
        int verifica = strcmp(palavra1, palavra2);
        if (verifica == 0){
            printf("São iguais.");
        }
        else{
            printf("São diferentes.");
        }
    }
}

int main()
{
    exemplo1();
    return 0;
}
```

Função strupr()

Sistemas para Internet
UFSM

- Converte todo o conteúdo da string para letras maiúsculas.

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

void exemploStrings()
{
    char a[20] = {"aula de algoritmos"};

    strupr(a);

    puts(a);

    printf("\n\n");

}
```



Função strupr()

Sistemas para Internet
UFSM

- Também pode ser usada para variáveis do tipo **char** com apenas um caractere, apenas acrescentando & antes da variável.

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

void exemploStrings()
{
    char s = {'m'};

    strupr(&s);

    printf("%c", s);

    printf("\n\n");

}
```

```
C:\Users\Leandro\Desktop\STRUPR exem
M

Press any key to continue .
```

Função strlwr()

Sistemas para Internet
UFSM

- Converte todo o conteúdo da string para letras minúsculas.

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

void exemploStrings()
{
     char a[20] = {"AULA DE ALGORITMOS"};

     strlwr(a);

     puts(a);

     printf("\n\n");

}
```

```
C:\Users\Leandro\Desktop\STRLWR exem
aula de algoritmos

Press any key to continue . . .
```

Função strlwr()

Sistemas para Internet
UFSM

- Também pode ser usada para variáveis do tipo **char** com apenas um caractere, apenas acrescentando & antes da variável.

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

void exemploStrings()
{
     char s = {'F'};

     strlwr(&s);

     printf("%c", s);

     printf("\n\n");

}
```

```
C:\Users\Leandro\Desktop\STRLWR exem
f

Press any key to continue . . .
```

Exercício 4

Sistemas para Internet
UFSM

- Aplique a função **strupr()** ou **strlwr()** no exercício 3 para garantir que palavras iguais não sejam consideradas distintas devido a letras maiúsculas ou minúsculas**.**

- Pesquise sobre outras funções da biblioteca “string.h” e crie algoritmos simples para testá-las.

Exercício 4

Sistemas para Internet
UFSM

- Solução:

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

void exemplo1(){
    char palavra1[20], palavra2[20];
    printf("digite a palavra 1: ");
    gets(palavra1);
    printf("digite a palavra 2: ");
    gets(palavra2);
    int tam_p1, tam_p2;
    tam_p1 = strlen(palavra1);
    tam_p2 = strlen(palavra2);
    if(tam_p1 == tam_p2){
        strlwr(palavra1);
        strlwr(palavra2);
        int verifica = strcmp(palavra1, palavra2);
        if (verifica == 0){
            printf("São iguais.");
        }
        else{
            printf("São diferentes.");
        }
    }
}

int main()
{
    exemplo1();
    return 0;
}
```