

Strings

C3

FURG - Fundação Universidade Federal de Rio Grande
C3 - Centro de Ciências Computacionais

Strings

- ▶ Em C um caractere é representado entre aspas simples. Ex.: 'a', 'b', '0', ...
- ▶ Uma cadeia de caracteres, mais conhecida como *string*, é uma sequência de letras e símbolos.
- ▶ Em C uma string é representada entre aspas duplas. Ex.: "palavra", "Isso eh uma frase", "a", " ", "", ...
- ▶ Onde os símbolos podem ser espaços em branco, dígitos e vários outros como pontos de exclamação e interrogação, símbolos matemáticos, etc.
- ▶ Em C, uma cadeia de caracteres é representada por um vetor de variáveis do tipo `char` e é sempre terminada com o caractere especial: `'\0'`.

Exemplo

```
/* Definindo e inicializando uma cadeia  
   de caracteres */  
char texto[100] = "Ola Mondo!";  
  
/* Como e' um vetor, podemos corrigir o  
   caractere errado da posicao 5 */  
texto[5] = 'u';
```

Lendo e Escrevendo Strings

A sequência especial %s deve ser usada nos comandos printf e scanf para mostrar ou ler uma string, respectivamente.

```
/* Para mostrar na tela todo o texto
   armazenado na variavel texto */
printf("%s", texto);

/* Para ler um texto do teclado
   (a sequencia %s aplicada ao scanf
   aceita somente uma palavra, sem espacos) */
scanf("%s", texto);
```

Note que para ler uma string usando scanf, não usamos o &.

Lendo Strings com Espaço

- ▶ Utiliza-se o comando `fgets`.

- ▶ Sintaxe:

`fgets(var, tam_maximo, canal_entrada)`

- ▶ Exemplos:

```
char frase[100];
```

```
fgets(frase, 100-1, stdin);
```

```
/* Note que:
```

- o tamanho deve ser
no maximo o tam da string - 1 (`'\0'`).
- o canal de entrada para ler do teclado
eh o `stdin` (standard input).
- o `fgets` guarda o `'\n'` (enter) na string lida.

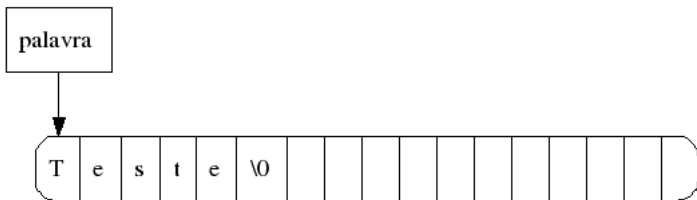
```
*/
```

Caracteres Especiais

- ▶ Dentro dos caracteres disponíveis em C, existem alguns caracteres especiais.
- ▶ Os principais são:
 - ▶ `'\n'` - Pula linha.
 - ▶ `'\t'` - Tab.
 - ▶ `'\0'` - Caractere nulo.

Caractere NULO

- ▶ O caractere nulo ou `'\0'` é extremamente importante na manipulação de strings.
- ▶ Toda a string deve ser encerrada com `'\0'`. Exemplo:



- ▶ CUIDADO: ao definir o tamanho de uma string sempre considere que o `'\0'` ocupa pelo menos uma posição.

Biblioteca string.h

As cadeias de caracteres são tão importantes que existe uma biblioteca de funções só com comandos para ela. É a biblioteca `string.h`.

Principais funções da biblioteca string.h

strlen(texto)	Retorna o tamanho da string texto em número de caracteres.
strcpy(destino, fonte)	Copia a string fonte para a string destino.
strcat(destino, fonte)	Concatena a string fonte no fim da string destino.
strcmp(str1, str2)	Compara strings. Retorna 0 (zero) caso elas sejam iguais, um número negativo caso str1 seja menor que str2 e um número positivo caso contrário.

Exemplo - strlen

- Programa que lê uma string e mostra a quantidade de caracteres da string.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char texto[TAMANHO];
    int tam;

    printf("Digite uma texto: ");
    fgets(texto, TAMANHO-1,stdin);

    // Calcula o tamanho da palavra
    tam = strlen(texto);

    printf("O texto que voce digitou tem %d caracteres. \n",tam);
    return 0;
}
```

Exercício - strlen

- ▶ Faça um programa que lê uma string e mostra a quantidade de caracteres da string e a quantidade de letras 'a' da string.

Exercício - strlen

- Faça um programa que lê uma string e mostra a quantidade de caracteres da string e a quantidade de letras 'a' da string.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char texto[TAMANHO];
    int tam,i;
    int cont=0;

    printf("Digite uma texto: ");
    fgets(texto, TAMANHO-1,stdin);

    // Calcula o tamanho da palavra
    tam = strlen(texto);

    for(i=0;i<tam;i++)
        if(texto[i]=='a')
            cont++;

    printf("O texto que voce digitou tem %d caracteres. \n",tam);
    printf("O texto que voce digitou tem %d letras 'a'. \n",cont);
    return 0;
}
```

Exemplo - strcpy

- Programa que lê uma string, faz uma cópia da string e mostra as duas strings na tela.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char original[TAMANHO];
    char copia[TAMANHO];

    printf("Digite uma texto: ");
    fgets(original, TAMANHO-1, stdin);

    // Copia a string original dentro de copia
    strcpy(copia, original); //Equivalente a: copia=original;

    printf("O texto que voce digitou foi: %s\n", original);
    printf("Copia do texto que voce digitou: %s\n", copia);
    return 0;
}
```

Exercício - strcpy

- ▶ Faça um programa que leia o nome e a idade de 20 pessoas e mostre o nome e a idade da pessoa mais velha.

Exercício - strcpy

- Faça um programa que leia o nome e a idade de 20 pessoas e mostre o nome e a idade da pessoa mais velha.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200
#define N 20

int main() {
    char nome[TAMANHO];
    int idade;
    char nomeMaisVelho[TAMANHO]="";
    int idadeMaisVelho=-1;
    int i;

    for(i=0;i<N;i++){
        printf("Digite o nome: ");
        fgets(nome, TAMANHO-1,stdin);
        printf("Digite a idade: ");
        scanf("%d",&idade);
        getchar(); //Remove lixo do buffer de entrada
        if(idade>idadeMaisVelho){
            idadeMaisVelho=idade;
            strcpy(nomeMaisVelho,nome);
        }
    }

    printf("Nome....: %s\n",nomeMaisVelho);
    printf("Idade....: %d\n",idadeMaisVelho);
    return 0;
}
```

Exemplo - strcat

- Programa que lê o sobrenome e o nome de uma pessoa e concatena em uma variável.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char nome[TAMANHO];
    char sobrenome[TAMANHO];

    printf("Digite o Sobrenome: ");
    scanf("%s", sobrenome);

    printf("Digite o Primeiro Nome: ");
    scanf("%s", nome);

    //Adiciona um espaco depois do nome
    strcat(nome, " ");
    //Adiciona o sobrenome ao final do nome e espaco.
    strcat(nome, sobrenome);

    printf("Nome....: %s\n", nome);

    return 0;
}
```


Exercício - strcat

- ▶ Faça um programa que transforma um número inteiro decimal em um número binário. Guarde o binário em uma string.

Exercício - strcat

- Faça um programa que transforma um número inteiro decimal em um número binário. Guarde o binário em uma string.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char bin[TAMANHO]="";
    char aux[TAMANHO]="";
    int dec;

    printf("Digite o nro em decimal: ");
    scanf("%d",&dec);

    while(dec>0){
        if (dec%2==0)
            strcpy(aux,"0");
        else
            strcpy(aux,"1");
        strcat(aux,bin);
        strcpy(bin,aux);
        dec=dec/2;
    }

    printf("Numero em Binario: %s\n",bin);

    return 0;
}
```

Exemplo - strcmp

- ▶ Programa que lê duas strings e verifica se elas são iguais.

```
#include <stdio.h>
#include <string.h>
#define TAMANHO 200

int main() {
    char texto01[TAMANHO];
    char texto02[TAMANHO];

    printf("Digite um texto: ");
    fgets(texto01, TAMANHO-1, stdin);

    printf("Digite outro texto: ");
    fgets(texto02, TAMANHO-1, stdin);

    if(strcmp(texto01, texto02) == 0)
        printf("Os textos digitados são iguais.\n");
    else
        printf("Os textos digitados são diferentes.\n");

    return 0;
}
```

Exercício - strcmp

- ▶ Faça um programa que fique lendo uma string até o usuário acertar uma senha pré-determinada (“password”).

Exercício - strcmp

- Faça um programa que fique lendo uma string até o usuário acertar uma senha pré-determinada (“password”).

```
/* exemplo strcmp */
#include <stdio.h>
#include <string.h>
#define TAM 80

int main () {
    char senha[TAM] = "password\n";
    char entrada[TAM];

    printf ("Qual a senha? ");
    fgets(entrada,TAM-1,stdin);

    while (strcmp (senha,entrada) != 0) {
        printf("Senha errada. Tente de novo.\n");
        printf ("Qual a senha? ");
        fgets(entrada,TAM-1,stdin);
    }

    printf ("Voce acertou a senha!\n");
    return 0;
}
```

Tabela ASCII

- ▶ Em C, todo caractere é representado por um valor entre 0 e 255 (1 byte).
- ▶ Cada caractere tem um código numérico correspondente.
- ▶ Essa tabela de relação entre símbolo e o seu respectivo código numérico é denominada tabela ASCII (*American Standard Code for Information Interchange*).
- ▶ Por exemplo:

'A' - 65	'a' - 97
'B' - 66	'b' - 98
'C' - 67	'c' - 99
...	
'[' - 91	'{' - 123

Exercício

- ▶ Faça um programa que leia uma string e que transforma todos os caracteres da string em maiúsculos.

Exercício

- Faça um programa que leia uma string e que transforma todos os caracteres da string em maiúsculos.

```
#include <stdio.h>
#include <string.h>
int main(){
    char s[100];

    printf("Digite uma palavra: ");
    fgets(s,100-1,stdin);

    for(i=0;i<strlen(s);i++)
        if(s[i]>='a' && s[i]<='z')
            s[i]=s[i]-32; // 32 dif entre 'a' e 'A'

    printf("%s\n",s);
    return 0;
}
```


Exercício

Faça um programa que leia 10 palavras e uma nova palavra e mostre a posição da nova palavra na lista de palavras anteriores. Caso a palavra não tenha sido digitada anteriormente, mostrar uma mensagem informando.

Resposta Exercício

```
#include <stdio.h>
#include <string.h>
int main(){
    char s[10][100]; // 10 Strings de 100 Caracteres cada.
    char palavra[100]; int k;

    for (k=0;k<10;k++) {
        printf("Digite uma palavra: ");
        fgets(s[k],100-1,stdin);
    }
    printf("Digite a palavra que deseja procurar: ");
    fgets(palavra,100-1,stdin);

    k=0;
    while ((k<10) && (strcmp(palavra,s[k])!=0)
        k++;

    /* Testa por qual das duas condicoes o while parou */
    if (strcmp(palavra, s[k]) != 0)
        printf ("Achei a palavra %s na posicao %d. \n", s[k],k);
    else
        printf("Nao foi encontrada a palavra desejada.\n");
    return 0;
}
```

Conversão De/Para strings

- ▶ Existem variações dos comandos `printf` e `scanf` que ao invés de escrever na tela ou ler do teclado, eles escrevem/lêem de uma string.
- ▶ Os comandos têm um `s` (de string) na frente, portanto, são os comandos `sprintf` e `sscanf`.
- ▶ Estas variações funcionam de forma similar ao `printf` e `scanf`, a única diferença é que recebem como primeiro argumento a string que vai ser escrita ou lida.
- ▶ Assim é possível converter um inteiro (ou float) para string. Basta mandar “imprimir” em uma string.
- ▶ E também é possível converter uma string em inteiro, basta ler de uma string.

Convertendo cadeias de caracteres em números e vice-versa

```
#include <stdio.h>

int main() {
    /* Declaramos algumas cadeias de caracteres */
    char v1[10], v2[10];

    /* Fazemos algumas contas */
    float f = 3.25 * 5.73;
    int i = 5*4*3*2*1;
    float f2;
    int i2;

    /* Conversoes de numeros para strings */

    sprintf(v1, "%.1f", f); /* escreve o valor de f para string v1 */
    printf("%s\n", v1);     /* mostra string v1 */

    sprintf(v2, "%d", i);   /* escreve o valor de i para string v2 */
    printf("%s\n", v2);     /* mostra string v2 */

    /* Conversoes de strings para numeros */

    sscanf(v1, "%f", &f2); /* le um float da string v1 para f2 */
    printf("%.2f\n", f2);   /* mostra float f2 (perdeu precisao) */

    sscanf(v2, "%d", &i2);  /* le um int da string v2 para i2 */
    printf("%d\n", i2);     /* mostra int i2 */

    return 0;
}
```

Referências

- ▶ ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da programação de computadores. Pearson Prentice Hall, 2007.
- ▶ Schildt, Herbert. C completo e total. Pearson Makron Books, 1997.
- ▶ Notas de aula do Prof. Flavio Keidi Miyazawa.
- ▶ Notas de aula do Prof. Emanuel Estrada.
- ▶ Notas de aula do Prof. Alessandro Bicho.