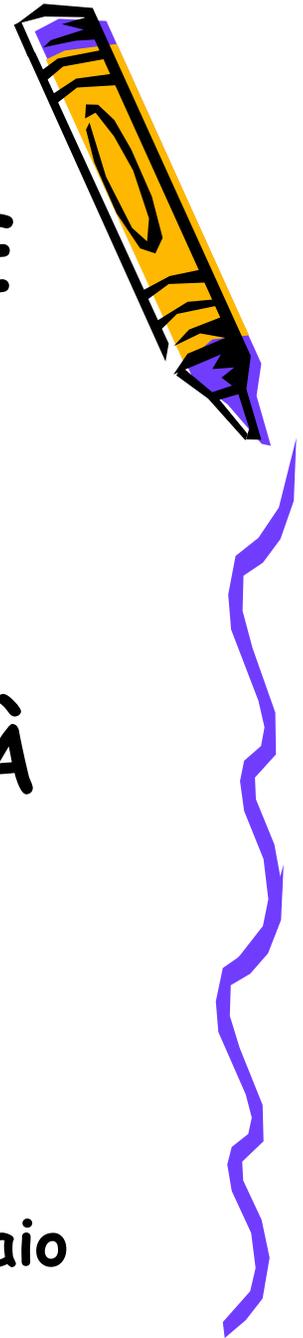
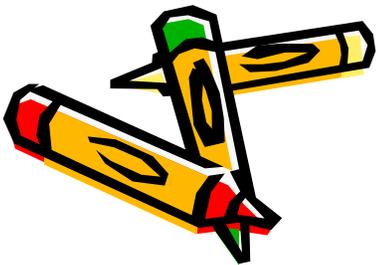


CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS



DISCIPLINA : INTRODUÇÃO À
LÓGICA DE PROGRAMAÇÃO

ASSUNTO: C



PROFESSOR : Romilson Lopes Sampaio

STRINGS

Não existe um tipo String em C;

Strings em C são uma array do tipo char. Devemos apenas ficar atentos para o fato de que as strings têm o seu último elemento como um '\0'.

A declaração geral para uma string é:

char nome_da_string [tamanho];

INICIALIZAÇÃO DE STRINGS

- Uma string pode ser inicializada na sua declaração com uma sequência de caracteres entre chaves e separadas por vírgula.

```
char vetc[6]= {'T', 'e', 'x', 't', 'o', '\0'};
```

- *Lembre-se que o compilador só reconhecerá um caractere se este estiver entre aspas simples, logo usar uma atribuição do tipo {t,e,x,t,o,\0} ou {texto\0} irá gerar um erro de compilação.*

INICIALIZAÇÃO DE STRINGS

- Uma string pode também ser inicializada por uma seqüência de caracteres entre aspas duplas. Neste caso, não é necessário o uso de aspas simples e virgulas, o compilador C coloca automaticamente o '\0' no final.

```
char vetc[6] = "Texto";
```

- Assim como vetores, na inicialização de uma string o seu tamanho pode ser omitido.
- O compilador vai verificar e considerar o tamanho declarado na inicialização.

```
char vetc[ ] = "Texto"; /* vetor não-dimensionado, o  
compilador coloca automaticamente  
o '\0' no final */
```

EXEMPLO

manipulação de string

```
#include <stdio.h>
#include <stdlib.h>
main ()
{
    char nome[] = "fulano";
    char sobrenome[] = "de tal";
    char nomeCompleto[] = nome + sobrenome;
    int i=0;
    while(nomeCompleto[i] != '\0'){
        printf("%c",nomeCompleto[i]);
        nomeCompleto[i++]; }
    system("PAUSE");
}
```

EXEMPLO

- *PORQUE O PROGRAMA ANTERIOR NÃO COMPILA?*
- Vamos entender o que o programa faz. Declara uma string contendo um nome, uma segunda string que é inicializada com o sobrenome, e uma terceira string que recebe a string nome concatenada a string sobrenome. Em seguida uma estrutura de fluxo while controla a impressão do conteúdo da string nomeCompleto.
- Um erro muito comum no uso de string em C esta sendo cometido na linha 7 do programa anterior:
char nomeCompleto[] = nome + sobrenome;
Strings não podem ser concatenadas utilizando o operador +. Existe uma diretiva em C que implementa diversas funções de manipulação de valores em string.

EXEMPLO

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main (){
char nome[] = "fulano";
char sobrenome[] = " de tal";
char nomeCompleto[15];
strcat(nome, sobrenome); strcpy(nomeCompleto,
    nome); puts(nomeCompleto);
system("pause"); }
```

FUNÇÕES PARA MANIPULAÇÃO DE STRINGS

gets

A função **gets()** lê uma string do teclado. Sua forma geral é:

```
gets (nome_da_string);
```

O programa abaixo demonstra o funcionamento da função **gets()**:

```
#include <stdio.h>
int main ()
{
char string[100];
printf ("Digite o seu nome: ");
gets (string);
printf ("\n\n Ola %s",string);
}
```

Repare que é válido passar para a função **printf()** o nome da string. Como o primeiro argumento da função **printf()** é uma string também é válido fazer:

`printf (string);` isto simplesmente imprimirá a string.

A função **gets()** coloca o terminador nulo na string, quando você aperta a tecla "Enter"

FUNÇÕES PARA MANIPULAÇÃO DE STRINGS

strcpy

Sua forma geral é:

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

A função **strcpy()** copia a string-origem para a string- destino.

EXEMPLO

```
#include <stdio.h>
#include <string.h>
int main ()
{
char str1[100],str2[100],str3[100];
printf ("Entre com uma string: ");
gets (str1);
strcpy (str2,str1); /* Copia str1 em str2 */
strcpy (str3,"Voce digitou a string "); /* Copia "Voce digitou a string" em str3
*/
printf ("\n\n%s%s",str3,str2);
}
```

FUNÇÕES PARA MANIPULAÇÃO DE STRINGS

strcat

A função **strcat()** tem a seguinte forma geral:

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

A string de origem permanecerá inalterada e será anexada ao fim da string de destino.

EXEMPLO

```
#include <stdio.h>
#include <string.h>
main ()
{
char str1[100],str2[100];
printf ("Entre com uma string: ");
gets (str1);
strcpy (str2,"Voce digitou a string ");
strcat (str2,str1); /* str2 armazenara' Voce digitou a string + o conteudo de
str1 */
printf ("\n\n%s",str2);
}
```

FUNÇÕES PARA MANIPULAÇÃO DE STRINGS

strlen

Sua forma geral é:

strlen (string);

A função **strlen()** retorna o comprimento da string fornecida. O terminador nulo não é contado. Isto quer dizer que, de fato, o comprimento do vetor da string deve ser um a mais que o inteiro retornado por **strlen()**.

EXEMPLO

```
#include <stdio.h>
#include <string.h>
main ()
{
int size; char str[100];
printf ("Entre com uma string: ");
gets (str);
size=strlen (str);
printf ("\n\nA string que voce digitou tem tamanho %d",size);
}
```

FUNÇÕES PARA MANIPULAÇÃO DE STRINGS

strcmp

Sua forma geral é:

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

A função **strcmp()** compara a string 1 com a string 2. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não-zero.

EXEMPLO

```
#include <stdio.h>
#include <string.h>
main ()
{
char str1[100],str2[100];
printf ("Entre com uma string: ");
gets (str1);
printf ("\n\nEntre com outra string: ");
gets (str2);
if (strcmp(str1,str2))
    printf ("\n\nAs duas strings são diferentes.");
else
    printf ("\n\nAs duas strings são iguais.");
}
```

Pode utilizar scanf no lugar do gets, só que o scanf não lê espaços em branco.

Se o usuário digitar:

>“Alexandre Costa e Silva”

o scanf vai pegar apenas “Alexandre”.

O gets pega tudo.

EXEMPLO DO USO DAS FUNÇÕES

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char palavra[30], palavra2[30], teste[30];
    int tam, comp;
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    printf ("A palavra digitada eh %s\n", palavra2);
    strcpy (teste,"aula teste");
    printf ("A palavra copiada eh %s\n", teste);
    strcat (teste, " 1");
    printf ("A palavra concatenada eh %s\n", teste);
    tam=strlen(palavra);
    printf ("\nO tamanho da primeira string eh %d", tam);
    comp=strcmp (palavra, palavra2);
    if (comp==0)
        printf ("\nSao iguais: %d", comp);
        else
            printf ("\nSao diferentes: %d", comp);
    printf ("\nMaiusculo: %s",strupr(palavra));
    printf ("\nMinusculo: %s\n",strlwr(palavra));
    system("pause");
}
```

EXEMPLO

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

main() {
    char s[256];
    for (int i=0; i<26; i++) {
        s[i] = 'A' + i;
    }
    printf("A string contém\n");
    printf("%s\n", s);
    system("pause");
}
```

EXERCÍCIOS

1. Escreva um programa que leia duas strings e as coloque na tela. Imprima também a segunda letra de cada string.

2. Escreva um programa que leia uma string, conte quantos caracteres desta string são iguais a 'a' e substitua os que forem iguais a 'a' por 'b'. O programa deve imprimir o número de caracteres modificados e a string modificada.

3. Faça um programa que leia o nome de 5 pessoas e mostre os nomes armazenados. Utilize vetores.

4. Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo. Faça um programa que leia o salário e o cargo de um funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá então receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

•Cargo	Percentual
•Gerente	10%
•Engenheiro	20%
•Técnico	30%

1.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char palavra[30];
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c",
palavra[1]);
    printf ("\n Informe uma segunda string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c\n",
palavra[1]);
    system("pause");
}
```

2.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char string[30];
    int tamanho, i, contaA=0;
    printf ("\nInforme uma string: ");
    gets (string);
    tamanho=strlen(string);
    for (i=0;i<tamanho;i++)    {
        if (string[i]=='a')    {
            contaA++;
            string[i]='b';
        }
    }
    printf ("\nO numero de caracteres modificados eh %d", contaA);
    printf ("\nA string modificada eh %s\n", string);
    system("pause");
}
```

3.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char nomes[5][10];
    int cont;
    for (cont=0;cont<5;cont++)
    {
        printf ("\nInforme o nome %d: ", cont);
        gets(nomes[cont]);
    }
    for (cont=0;cont<5;cont++)
        printf ("\nO nome armazenado na posicao %d eh %s",
cont, nomes[cont]);
    system("pause");
}
```

4.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(){
    char cargo[20];
    float salario, novoSalario;
    printf ("\nDigite o cargo: ");
    scanf ("%s", cargo);
    printf ("\nDigite o salario: ");
    scanf ("%f", &salario);
    if (strcmp(cargo, "gerente")==0)
        novoSalario=salario*1.1;
    else if (strcmp(cargo, "engenheiro")==0)
        novoSalario=salario*1.2;
    else if (strcmp(cargo, "tecnico")==0)
        novoSalario=salario*1.3;
    else
        novoSalario=salario*1.4;
    printf ("\nO salario antigo eh %.2f, o salario novo eh %.2f e a diferenca eh
de R$ %.2f\n", salario, novoSalario, novoSalario-salario);
    system("pause");
}
```

EXERCÍCIOS

- 1. Faça um programa que carregue e imprima um vetor com o seu nome (nome completo) e o total de letras que ele possui.**
- 2. Faça um programa que carregue o vetor nome com o seu nome (nome completo), fornecido via teclado. Utilize a função gets para a leitura e em seguida use a função puts para imprimir o nome. Substitua a função gets pela função scanf e avalie os resultados.**
- 3. Faça um programa que verifique se uma palavra armazenada em um vetor de caracteres é um palíndromo.**
- 4. Faça um programa que carregue um vetor de caracteres e imprima o mesmo vetor com as letras minúsculas substituídas por letras maiúsculas.**
- 5. Faça um programa que carregue um vetor de caracteres, gere e imprima um outro vetor onde as vogais, do primeiro vetor, sejam substituídas pelo caracter *.**

1.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main ()
{
    char nome[50];
    int i,tam,cont;
    printf ("\nDigite seu nome completo.\n");
    gets(nome);
    cont=tam=strlen(nome);
    for (i=0;i<=tam;i++)
        if (nome[i]==' ') cont--;
    printf ("\n%d letras(s).\n\n",cont);
    system("pause");
}
```

2.

```
# include <stdio.h>
# include <string.h>
# include <stdlib.h>
main ()
{
    char nome[50];
    printf ("\nDigite seu nome completo.\n");
    gets(nome);
    puts(nome);
    printf ("\nReDigite seu nome completo.\n");
    scanf ("%s",&nome);
    puts(nome);
    system("pause");
}
```

3.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
main ()
{
    char palavra[15], inversa[15];
    printf ("\nDigite uma palavra.");
    gets(palavra);
    strcpy(inversa,palavra);
    strrev(inversa);
    if (!strcmp(palavra,inversa))
        printf ("\nA PALAVRA %s E UM PALINDROMO.\n\n",palavra);
    else
    {
        printf ("\nA PALAVRA %s NAO E UM
PALINDROMO.\n\n",palavra);
        puts(inversa);
    }
    system("pause");
}
```

4.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
main ()
{
    char vet[30];
    int i;
    printf ("\nDIGITE LETRAS MAISCULAS E MINUSCULAS
    ALEATORIAMENTE.\n");
    gets(vet);
    printf ("\nVETOR MODIFICADO.\n\n");
    for (i=0;i<30;i++)
    {
        if (islower(vet[i]))
        {
            vet[i]=toupper(vet[i]);
        }
        printf ("%c",vet[i]);
    }
    system("pause");
}
```

5.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main ()
{
    char v1[20],v2[20];
    int tam,i=0;
    printf ("\nDIGITE UMA FRASE.");
    gets(v1);
    tam=strlen(v1);
    for (i=0;i<=tam;i++)
    {
        if ((v1[i]=='a')||(v1[i]=='e')||(v1[i]=='i')||(v1[i]=='o')||(v1[i]=='u'))
            v2[i]='*';
        else
            v2[i]=v1[i];
    }
    printf ("\nFRASE MODIFICADA\n\n");
    puts(v2);
    printf ("\n\n");
    system("pause");
}
```