

Vetores e Strings

4. Funções Básicas para manipulação de Strings

- **gets**

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

```
gets (nome_da_string);
```

Vetores e Strings

- gets (continuação)

Exemplo:

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

Vetores e Strings

4. Funções Básicas para manipulação de Strings (continuação)

- **strcpy**

Sua forma geral é:

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

A função **strcpy()** copia o conteúdo da `string_origem` para a `string_destino`. As funções para manipulação de strings apresentadas neste tópico estão no arquivo cabeçalho **string.h**.

Vetores e Strings

- strcpy (continuação)

Exemplo:

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

Vetores e Strings

4. Funções Básicas para manipulação de Strings (continuação)

- **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()**.

Vetores e Strings

Exemplo:

```
#include <stdio.h>
#include <string.h>
main ()
{
    int size;
    char str[100];
    printf ("Entre com uma string: ");
    gets (str);
    size=strlen (str);
    printf ("\nA string que voce digitou tem tamanho
%d", size+1);
}
```

Vetores e Strings

4. Funções Básicas para manipulação de Strings (continuação)

- **strcat**

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

strcat (string_destino, string_origem);

A função `strcat()` concatena a *string_destino* com a *string_origem*. A *string_origem* permanecerá inalterada e será anexada ao fim da *string_destino*.

Vetores e Strings

- strcat (continuação)

Exemplo:

```
#include <stdio.h>
#include <string.h>
main ()
{
    char str1[50],str2[100];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2,"Voce digitou a string ");
    strcat (str2,str1);
    printf ("\n\n%s",str2);
}
```

Vetores e Strings

4. Funções Básicas para manipulação de Strings (continuação)

- **strcmp**

Sua forma geral é:

strcmp (string1, string2);

A função `strcmp()` compara a `string1` com a `string2`. Se as duas forem idênticas a função retorna zero. Se elas forem diferentes a função retorna não-zero.

Vetores e Strings

```
#include <stdio.h> /* Exemplo strcmp*/
#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 ("\nAs duas strings são diferentes!");
    else
        printf ("\nAs duas strings são iguais!");
}
```

Vetores e Strings

Exercício:

Construa um programa que leia duas strings fornecidas pelo usuário, através da entrada padrão, verifique se estas possuem o mesmo tamanho, caso possuam, as compare. Se forem iguais, retorne uma mensagem na saída padrão indicando este fato. Caso contrário, concatene-as e retorne o resultado desta operação na saída padrão.

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings

- **sprintf e sscanf**

sprintf e sscanf são semelhantes a printf e scanf. Porém, ao invés de escreverem na saída padrão ou lerem da entrada padrão, escrevem ou lêem em uma string. Suas formas gerais são:

sprintf (string_destino,string_de_controle,lista_de_argumentos);

sscanf (string_origem,string_de_controle,lista_de_argumentos);

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings (continuação)

Estas funções são muito utilizadas para fazer a conversão entre dados na forma numérica e sua representação na forma de strings e vice-versa. No programa a seguir, por exemplo, a variável `i` é "impressa" em `string1`. Além da representação de `i` como uma string, `string1` também conterá "Valor de `i`=" .

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings (continuação)

Exemplo:

```
#include <stdio.h>
main()
{
    int i;
    char string1[25];
    printf( "Entre um valor inteiro: ");
    scanf("%d", &i);
    sprintf(string1, "Valor de i = %d", i);
    puts(string1);
```

201 }

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings (continuação)

Já no próximo programa, foi utilizada a função `sscanf` para converter a informação armazenada em `string1` em seu valor numérico:

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings (continuação)

Exemplo:

```
#include <stdio.h>
main()
{
    int i, j;
    float k;
    char string1[] = "10 20 5.89";
    sscanf(string1, "%d %d %f", &i, &j, &k);
    printf("Valores lidos: %d, %d, %.2f", i, j, k);
}
```

Vetores e Strings

4. Comandos de entrada e saída formatada que trabalham sobre strings (continuação)

Exercício:

Construa um programa que declare um vetor de strings e outro de inteiros, ambos com 10 elementos. O programa deve inicializar os vetores com valores fornecidos pelo usuário através da entrada padrão. Depois, acrescente o inteiro, do vetor de inteiros, no final da string correspondente no vetor de strings. Ao término do processamento o vetor de strings, com seus valores atualizados, deve ser impresso na saída padrão.