Computação 2

Aula 3

Vetores de caracteres (strings)

Profª. Fabiany
fabianyl@utfpr.edu.br

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Vetor de caracteres (Strings)

- Uma cadeia de caracteres, mais conhecida como string, é uma seqüência de caracteres (letras e símbolos) utilizada para o armazenamento de texto.
- Na linguagem C, strings são vetores de caracteres que possuem um caracter que indica o término de seu conteúdo, o caracter nulo ‘\0’ (contrabarra zero).
- Uma cadeia de caracteres é representada por um vetor de variáveis do tipo char.

Declaração de strings

- Como a string possui o caractere nulo para delimitar o final do seu conteúdo, o tamanho da string deve ser definido com um caractere a mais do que será efetivamente necessário.
- Sintaxe:

  char nome_variavel [tamanho+1];

- Exemplo.

  char vetch [10];

  vetch é um vetor de caracteres (string) de tamanho 10. Pode receber uma palavra de no máximo 9 letras.

Inicialização de strings

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

```
char letras[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 do C coloca automaticamente o '\0' no final.

char letras[6] = “Texto”;

- Assim como vetores e matrizes, 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.

Lendo strings do teclado

- Podemos ler uma cadeia de caracter ou string inteira, utilizando o formato %s.

scanf ("%s", letras); ou scanf (“%s”,&letras[0]);

- Neste caso não é necessário o e comercial (&) para strings. Isso ocorre pois o nome de um vetor já é um endereço de memória (o endereço de memória do começo do vetor).
- A leitura a partir do teclado utilizando o scanf é somente até o primeiro espaço, ou seja, lê somente uma palavra.

Lendo strings do teclado

Outra formas de entrada:

- **gets(s)** - Lê uma string do dispositivo de entrada padrão e armazena esta string em s.

- **fgets(s, TAM, stdin)** - Lê uma string de tamanho TAM do dispositivo de entrada padrão e armazena esta string em s.

Escrevendo strings

- Para escrever uma cadeia de caracteres (string) usamos a função printf com o formato %s.

```
printf ("%s", letras);
```

- Da mesma forma do gets e fgets, temos o puts e fputs, que escrevem a string na tela.

```
puts (letras);
fputs (letras, stdout);
```

Manipulando strings

- A biblioteca padrão da linguagem C fornece várias funções úteis para manipular strings.Para usá-las, você deve incluir o cabeçalho string.h no início dos seus arquivos.
- ✓ **strlen**(s): Retorna o tamanho da cadeia texto em número de caracteres.
- ✓ **strcpy**(destino, fonte): Copia a cadeia fonte para a cadeia destino.
- ✓ **strcat**(destino, fonte): Concatena a cadeia fonte no fim da cadeia destino.

Manipulando strings

- **Exemplo: strlen(s):**

  char nome[15] = "Maria da Silva";

  int s = strlen (nome); // *s* conterá o valor 14

- **Exemplo : strcpy(destino,origem):**

  char nome[50] ;

  char nome2[] = “Homer Simpson";

  strcpy (nome, nome2); // agora *nome* conterá “Homer Simpson"

- **Exemplo: strcat(destino,fonte)**:

  char nome[] = “Homer";

  char sobrenome[] = “ Simpson";

  strcat (nome, sobrenome); // *nome* vai ser “Homer Simpson”

Manipulando strings

- Para comparar *o conteúdo* de duas strings, deve-se usar a função:

int **strcmp** (char **s1*, char **s2*);

- Compara duas cadeias de caracteres e retorna um valor:
  - ✓ 0: se s1 e s2 forem iguais;
  - ✓ < 0: se s1 for menor que s2;
  - ✓ > 0: se s1 for maior que s2.

Manipulando strings

Exemplo: **strcmp** (string1,string2);

```
void main ()
{
    char nome[100], nome2[100];
    int n;

    printf ( "Digite um nome:\n");
    gets(nome);

   printf ( "Digite outro nome:\n");
   gets(nome2);

  n = strcmp(nome, nome2);

  if(n == 0)  //ou if(strcmp(nome1,nome2)==0)
   printf("Nomes iguais\n");
  else
   printf("Nomes diferentes\n");

}
```

Exercícios

1. Faça um programa que dada uma string (cadeia de caracteres) pelo teclado, o programa deve imprimir "***" toda a vez que aparecer um "a" na mensagem original. Ex: se o usuário digitou “arara" será impresso "***r***r***" na tela.

2. Elabore um programa que leia uma string e conte quantas vogais há nela.

3. Faça um programa que contabilize o nº de ocorrências de cada uma das 26 letras em uma mensagem entrada pelo usuário.
   Ex.: "ola, como vai voce?" Vai resultar: a:2, b:0, c:2, d:0, e:1, ..., p:0, o:4, ..., z:0.