Estruturas homogêneas

Strings de caracteres

Prof. Fabio Takeshi Matsunaga

5 de junho de 2019

Vetores

- Um vetor é uma coleção de variáveis do mesmo tipo referenciada por um nome comum;
- As coleções de variáveis são armazenadas em posições contíguas da memória;
- O acesso a um elemento específico é feitor por um índice;
- Em linguagem C, os índices iniciam na posição 0;

Strings

- Uma das aplicações mais comuns de vetores é na representação de palavras e dados textuais;
- Palavras e textos são conhecidas como *strings* no mundo da programação;
- Uma *string* é um vetor de caracteres (do tipo `char`);

Declaração de uma *string*

```
1 int main() {
2   char palavra[11];
3 }
```

O programa definiu uma string que guarda 10 caracteres. É recomendável sempre reservar espaço na memória com uma quantidade mais longa do que a palavra irá armazenar.

Declaração de uma *string*

```
1  int main() {
2   char hello[7]={'H','e','l','l','o','!','\0'};
3
4   printf("%s",hello);
5
6   printf("\n");
7   system("pause");
8  }
```

- Neste programa, os caracteres da string já foram inicializadas.
- Toda string deve ser finalizada com um caracter nulo (\0), para indicar o final da string.
- O caracter de formatação para string é o %s.

Declaração de uma *string*

```
1  int main() {
2    char nome[50];
3
4    printf("Qual o seu nome? ");
5    scanf("%s",nome);
6
7    printf("Ola %s! Seja muito bem vindo! :)\n",
         nome);
8    system("pause");
9  }
```

- Para ler uma string, utiliza-se %s no `scanf`. Entretanto, para string, não é necessário utilizar o &, visto que um vetor já é um endereço de uma variável.
- Não é necessário se preocupar em adicionar o caracter nulo no final da string, pois a função de leitura já faz isso automaticamente.

Biblioteca <string.h>

- A linguagem C suporta diversas funções e comandos para manipulação de strings.
- Estas funções estão contidas na biblioteca `string.h`.

Leitura e escrita

- Leitura (`gets(s1)`) – equivale ao `scanf` de uma string.
- Escrita (`puts(s1)`) – equivale ao `printf` de uma string.

```
1  char s1[80], s2[80];
2
3  puts("Digite s1");
4  gets(s1);
5
6  puts("Digite s2");
7  gets(s1);
```

Cópia de string (atribuição)

- `strcpy(s1,s2)`.
- Copia o valor da string s2 em s1.

```
1 strcpy(s3,s1);
2 printf("Valor de s3: %s\n",s3);
```

Tamanho da string

- `strlen(s1)`.
- Retorna a quantidade de caracteres contida na string (sem contar o caracter nulo).
- O valor retornado por essa função é um inteiro.

```
1 comp1 = strlen(s1);
2 comp2 = strlen(s2);
3 printf("Tamanhos: s1=%d s2=%d\n",comp1,comp2);
```

Comparação entre strings

- `strcmp(s1,s2)`.
- Compara as duas strings.
- Se s1 e s2 forem iguais, retorna 0. Caso contrário, retorna um número menor que 0 se $s1 < s2$ e um número maior que 0 se $s1 > s2$.
- Como a função retorna falso se as duas strings forem iguais, é necessário utilizar o operador ! para reverter a condição.

```
1 if (!strcmp(s1,s2))
2         puts("As strings sao iguais");
3 else
4         puts("As strings sao diferentes");
```

Concatenação

- `strcat(s1,s2)`.
- Concatena s2 ao final de s1, isto é, faz uma junção de duas strings.

```
1 strcat(s3,s2);
2 printf("Valor de s3: %s\n",s3);
```

Considerações finais sobre strings

- Strings são vetores de caracteres;
- Para manipular strings, basta utilizar os mesmos princípios dos vetores – acesso a um caracter através de seu índice;
- Existem funções e comandos em linguagem C para manipular strings – biblioteca `string.h`.

Exemplo 1

- Faça um algoritmo que leia uma string e em seguida conte quantas letras 'a' existem na string.

Exemplo 2

Escreva um algoritmo que leia uma palavra (string). Em seguida, troque o primeiro caracter com o último, o segundo com o penúltimo, o terceiro com o antepenúltimo, e assim sucessivamente. Mostre a nova string depois da troca.

Exemplo 3

- Faça um algoritmo que leia uma string e verifique se a palavra é palíndroma ou não. Uma palavra palíndroma é lida, indiferentemente, da esquerda para a direita ou vice-versa. Exemplo: arara, aba, raiar, reviver.

Exemplo 4

Escreva um algoritmo que leia uma string e em seguida converta as letras minúsculas para maiúsculas. Faça também o inverso.