Manipulação de Strings

Em C, uma string é um conjunto de caracteres armazenados num vetor. As strings são representadas utilizando aspas enquanto os caracteres entre apóstrofes. A declaração é dessa forma:

```
char nome[30];
```

Para ler um string, podemos usar o scanf(), porém a leitura não irá aceitar espaços entre textos. Por isso, o ideal é usar a função gets(), assim:

```
printf("Digite nome: ");
gets(nome);
```

Obs: o gets() não precisa do fflush().

Para mostrar, se utiliza o especificador %s, assim:

```
printf("O nome é: %s\n", nome);
```

A declaração de strings obedece à sintaxe de declaração de vetores, sendo a primeira posição 0 e o final da string identificado pelo caracter '\0'. Isso é importante, pois como já foi visto, o C não faz tratamento de final de vetor, tendo, por isso, que delimitar o final de uma string. E isso é feito com o '\0'. A leitura já insere o caracter '\0'.

Definitivamente, o forte do C não é o tratamento direto de strings, visto que strings não são tipos básicos da linguagem e a única forma de representar um conjunto de caracteres é recorrendo a um vetor.

Por exemplo não podemos fazer:

- uma atribuição s1="OLA";
- uma comparação if (s1=="OLA") ...

Uma exceção é no momento da declaração, onde é possível realizar uma atribuição de valor:

```
char nome[10] = "OLA";
```

No entanto, C possui uma biblioteca de funções (string.h) que permite realizar estas tarefas e muito mais, e se por acaso não encontrarmos nenhuma função que faça o que queremos podemos sempre fazer manipulação caracter a caracter e resolver o problema.

Observe o exemplo a seguir. É possível fazer a atribuição utilizando uma função específica (strcpy()), ou simplesmente adicionando caracter a caracter nas posições adequadas, não esquecendo do '\0'.

```
#include <stdio.h>
#include <string.h>
void main()
{
	char string1[10], string2[10];
	printf("\nAtribuição de valor a strings\n");
	printf("----------------------------\n");
	strcpy(string1,"Ola");
	printf("\nAtribuicao na primeira forma: %s",string1);
	string2[0] = 'O';
	string2[1] = 'l';
```

```
	string2[2] = 'a';
	string2[3] = '\0';
	printf("\nAtribuicao na segunda forma: %s",string2);
}
```

Se o strcpy() é responsável pela atribuição de valor, sempre que quisermos comparar valores temos que usar o strcmp()ou o stricmp(). Esse segundo ignora caixa de texto na comparação. É importante destacar que a comparação feita entre dois strings pode retornar um valor igual, maior ou menor que zero:

- Se os strings forem iguais a função retorna 0
- Se o primeiro string for menor que o segundo, a função retorna um valor menor que 0
- Se o primeiro string for maior que o segundo, a função retorna um valor maior que 0

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

void main()
{
	char string1[20],string2[20];
	int retorno;

	printf("\nEntre com a primeira string: ");
	gets(string1);
	printf("\nEntre com a segunda string: ");
	gets(string2);

	retorno = strcmp(string1,string2);
	if(retorno == 0)
		printf("As strings sao iguais.\n");
	else if(retorno < 0)
		printf("A string1 e' menor.\n");
	else
		printf("A string2 e' menor.\n");

	retorno = stricmp(string1,string2);
	if(retorno == 0)
		printf("As strings sao iguais, desconsiderando caixa de texto\n");
	else if(retorno < 0)
		printf("A string1 e' menor, desconsiderando caixa de texto\n");
	else
		printf("A string2 e' menor, desconsiderando caixa de texto\n");
}
```

Em algumas situações é necessário saber o tamanho do string, que não é o tamanho do vetor, mas sim até onde se encontra o "\0". Para isso, basta usar o strlen(), assim:

```
	int tamanho = strlen(texto);
```

Isso significa que há duas maneiras de percorrer um vetor caracter a caracter:

```
int i;
for (i=0; i < strlen(texto); i++)
	printf("%c",texto[i]);
```

Ou:

```
int i;
for (i=0; texto[i] != '\0'; i++)
        printf("%c",texto[i]);
```

Verificando caracteres

Existem várias funções para manipular caracteres. Algumas delas podem ser vistas a seguir, todas ela definidas na biblioteca ctype.h.

Converte o caracter para maiúsculo:
`maiusculo = toupper(caracter);`

Converte o caracter para minúsculo:
`minusculo = tolower(int caracter);`

Verifica se é maiúsculo:
`if (isupper(caracter) == 1)`
Retorna true (1) se o caracter é maiúsculo e false (0) caso contrário.

Verifica se é maiúsculo:
`if (islower(caracter) == 1)`
Retorna true (1) se o caracter é minúsculo e false (0) caso contrário.

Verifica se é um caracter alfabético:
`if (isalpha(caracter) == 1)`
Retorna true (1) se o caracter é uma letra e false (0) caso contrário.

Verifica se é o caracter é um dígito:
`if (isdigit(caracter) == 1)`
Retorna true (1) se o caracter é um número e false (0) caso contrário.

EXERCÍCIOS:

1) Fazer um programa que, dadas 2 palavras, determine:
   a) Se as palavras são iguais.
   b) Se as palavras são iguais, independente de caixa de texto.
   c) Caso as palavras sejam diferentes (tanto no a. quanto no b.), qual delas tem maior comprimento.

2) Criar um programa que leia um string aluno1. Após, declare um string aluno2 e jogue o conteúdo de aluno1 em aluno2. Após, converta cada caracter de aluno1 para maiúsculo, mostrando os dois strings ao final.

3) Elaborar um programa que leia uma frase e armazene-a em um vetor de caracteres. Depois, verifique e mostre o número de espaços em branco na frase, o número de vogais, o número de consoantes e o número de dígitos.

4) Fazer um programa que, dado um nome completo, informe a abreviatura deste nome. Não se devem abreviar as preposições como: do, de, etc. A abreviatura deve vir separada por pontos. Ex: Paulo Jose de Alma Prado. Abreviatura: P.J.A.P.
5) Escreva um programa que leia uma palavra e verifique se ela é um palíndromo. Palíndromo é a palavra cuja leitura é a mesma, quer se faça da direita para a esquerda, quer da esquerda para a direita. Exemplo: ovo, anilina