

Exercícios

29) Com base no que foi estudado, construa um programa em C que leia duas *strings*, fornecidas pelo usuário através da entrada padrão, e verifique se a segunda *string* lida está contida no final da primeira, retornando na saída padrão o resultado da verificação.

```
#include <stdio.h>
#include <string.h>
#define tamanho 100
main ()
{
    char str1[tamanho],str2[tamanho],logico=1;
    int tam_str1, tam_str2;
    printf ("\nEntre com a primeira string: ");
    gets (str1);
    printf ("\nEntre com a segunda string: ");
    gets (str2);
    tam_str1=strlen(str1);/*função que retorna o n. de caracteres da string*/
    /*tam_str2=strlen(str2);*/
    for (tam_str2=0; str2[tam_str2]; tam_str2++);
    if (tam_str2>tam_str1)
        logico=0;
    else
```

```
for (;tam_str2;tam_str1--,tam_str2--)
    if (str1[tam_str1-1]!=str2[tam_str2-1])
    {
        logico=0;
        break;
    }
if (logico)
{
    printf ("\nA segunda string esta contida no final");
    printf (" da primeira!");
}
else
{
    printf ("\nA segunda string nao esta contida no ");
    printf ("final da primeira!");
}
}
```

```
#include <stdio.h>
#include <string.h>
#define tamanho 100
main ()
{
    char str1[tamanho],str2[tamanho];
    printf ("\nEntre com a primeira string: ");
    gets (str1);
    printf ("\nEntre com a segunda string: ");
    gets (str2);
    if (strlen(str1)>=strlen(str2) &&
        !strcmp(&str1[strlen(str1)-strlen(str2)],str2)) {
        printf ("\nA segunda string esta contida no final da");
        printf (" primeira!");
    }else{
        printf ("\nA segunda string nao esta contida no ");
        printf ("final da primeira!");
    }
}
```

Exercícios – Aula Prática XIII

Exercícios

31) Elabore o código fonte de um programa na linguagem C que leia uma string fornecida pelo usuário e a exiba na saída padrão.

Obs.: Conforme vimos, a linguagem C ao efetuar a leitura de uma string com a função *gets()* não podemos controlar o número máximo de caracteres que serão lidos e armazenados no vetor de caracteres utilizado para conter a string, o que pode gerar uma indexação inválida. Para solucionar este problema podemos usar a função *scanf()*, utilizando o código `%s` e determinando entre o caractere `'%'` e o `'s'` o número máximo de caracteres a serem lidos, contudo o *scanf()* finaliza a leitura da *string* ao encontrar um espaço.

Com base no que foi exposto elabore um programa que controle o número máximo de caracteres que serão lidos e garanta que este número não excederá o valor dez, contudo permita que o programa leia strings que contenham o caractere `' '` (espaço).

```
#include <stdio.h>
main()
{
    char string[11];
    printf("\nForneca uma string com no maximo 10");
    printf(" caracteres: ");
    scanf ("%10s", string);
    printf("\n\nA string fornecida foi: %s\n\n", string);
}
```

```
#include <stdio.h>
main()
{
    char string[11];
    printf("\nForneca uma string com no maximo 10");
    printf(" caracteres: ");
    gets (string);
    printf("\n\nA string fornecida foi: %s\n\n", string);
}
```

Exercícios

32) Implemente um programa, na linguagem C, que declare duas matrizes de inteiros uma com dimensão 2×3 e outra 3×2 . O programa deve as inicializar com valores fornecidos pelo usuário, através da entrada padrão, e efetuar a multiplicação das duas matrizes, retornando a matriz resultante, com layout apropriado, na saída padrão.