

5 8	→	Entrada: dimensão (linha e coluna) da matriz
2 f s 8 j m b B	→	Entrada: valores da 1ª linha da matriz
e c i o s u E a	→	Entrada: valores da 2ª linha da matriz
o u o i d s m k	→	Entrada: valores da 3ª linha da matriz
i t w 7 % o i m	→	Entrada: valores da 4ª linha da matriz
i l k * 8 r v N	→	Entrada: valores da 5ª linha da matriz
2 i o	→	Entrada: número de caracteres do padrão e o padrão
4	→	Saída: número de ocorrências do padrão

### Observações Importantes:

- Se logo no início, quando o usuário informar as dimensões da matriz e estas forem maiores do que 100 por 100, interrompa o procedimento e apresente uma saída com uma dupla de zeros espaçados ("0 0").
- É **obrigado** o uso de uma função chamada pela função `main()` que faça a busca na matriz e retorne o número de ocorrências do padrão nela. Esta função tem como parâmetros:
  - um ponteiro para o primeiro caractere da matriz (**obrigado** usá-lo para buscar na matriz);
  - o tamanho dessa matriz;
  - um ponteiro para o primeiro caractere do padrão a ser procurado;
  - o tamanho deste padrão;
- Use outros parâmetros para a função se julgar necessário ou até mesmo crie novas funções se precisar.

### Manipulação de Cadeias de Caracteres (string)

- Em C não existe um tipo de dado `string`, sendo usado em seu lugar uma matriz de caracteres.
- Uma `string` é um vetor de caracteres (`char`) que termina com `\0`.
  - por essa razão, deve ter sempre uma posição a mais do que o número de caracteres que se deseja.

### gets()

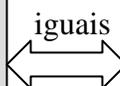
- Leitura de caracteres do teclado até que o ENTER seja pressionado.
- Não testa limites na matriz em que é chamada.
- Sintaxe: `gets( <nome da string> );`

```
using gets

#include <stdio.h>

int main()
{
    char str[80];

    gets(str);
    printf("\n%s", str);
    return 0;
}
```



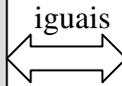
```
using scanf e índice de vetor

#include <stdio.h>
int main() {
    int i = 0;
    char str[80];
    do {
        scanf("%c", &str[i]);
        i++;
    } while ( str[i - 1] != '\n' );
    str[i - 1] = '\0';
    printf("\n%s", str);
    return 0;
}
```

## puts()

- Escrita de caracteres de uma string no monitor inserindo um \n no final.
- Reconhece códigos de barra invertida.
- Sintaxe: puts( <nome da string> );

usando puts
<pre>#include &lt;stdio.h&gt;  int main() {     char str[6] = "teste";      puts(str);     return 0; }</pre>

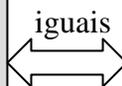


usando printf com código \s
<pre>#include &lt;stdio.h&gt;  int main() {     char str[6] = "teste";      printf("%s\n", str);     return 0; }</pre>

## strcat()

- Concatena duas strings e armazena o resultado na primeira (Obs.: não verifica tamanho das strings).
- Sintaxe: strcat( <1ª string>, <2ª string> );
- É necessário incluir a biblioteca string.h

usando strcat
<pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt;  int main() {     char str1[10] = "Bom";     char str2[10] = " dia";      strcat(str1, str2);     puts(str1);      return 0; }</pre>



usando atribuição a cada índice de vetor
<pre>#include &lt;stdio.h&gt;  int main() {     char str1[10] = "Bom";     char str2[10] = " dia";     int i1, i2 = 0;      for (i1 = 0; str1[i1] != '\0'; i1++) {     }      while (str2[i2] != '\0') {         str1[i1] = str2[i2];         i1++;         i2++;     }     str1[i1] = '\0';     puts(str1);      return 0; }</pre>

## strcmp()

- Compara duas strings e retorna 0 (zero) se elas forem iguais, 1 se a primeira for maior lexicograficamente que a segunda e -1 caso contrário.
- Sintaxe: `strcmp( <1ª string>, <2ª string> );`
- É necessário incluir a biblioteca `string.h`

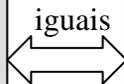
```
using namespace std;

usando strcmp

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

int main()
{
    char s1[6] = "teste";
    char s2[6] = "teste";

    printf("%d", strcmp(s1, s2));
    return 0;
}
```



```
using namespace std;

usando comparação a cada índice de vetor

#include <stdio.h>
int main()
{
    char s1[6] = "teste";
    char s2[6] = "teste";
    int i = 0;
    int iguais = 0;
    while(s1[i] != '\0' && s2[i] != '\0'){
        if (s1[i] > s2[i])
            iguais = 1;
        else if (s1[i] < s2[i])
            iguais = -1;
        i++;
    }
    if(iguais==0 && s1[i]!='\0' && s2[i]!='\0')
        iguais = -1;
    else if(iguais==0 && s1[i]!='\0' && s2[i]=='\0')
        iguais = 1;
    printf("%d", iguais);
    return 0;
}
```

## strcpy()

- Copia o conteúdo de uma string em outra.
- Sintaxe: `strcpy( <string destino>, <string origem> );`
- É necessário incluir a biblioteca `string.h`

```
using namespace std;

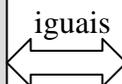
usando strcpy

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

int main()
{
    char str_orig[6] = "teste";
    char str_dest[6];

    strcpy(str_dest, str_orig);
    printf("\n%s", str_orig);
    printf("\n%s", str_dest);

    return 0;
}
```



```
using namespace std;

usando atribuição a cada índice de vetor

#include <stdio.h>

int main()
{
    char str_orig[6] = "teste";
    char str_dest[6];
    int i = 0;

    while (str_orig[i] != '\0') {
        str_dest[i] = str_orig[i];
        i++;
    }
    str_dest[i] = '\0';
    printf("\n%s", str_orig);
    printf("\n%s", str_dest);
    return 0;
}
```