Laboratório de Programação

Profa. Ms. Valéria Pinheiro

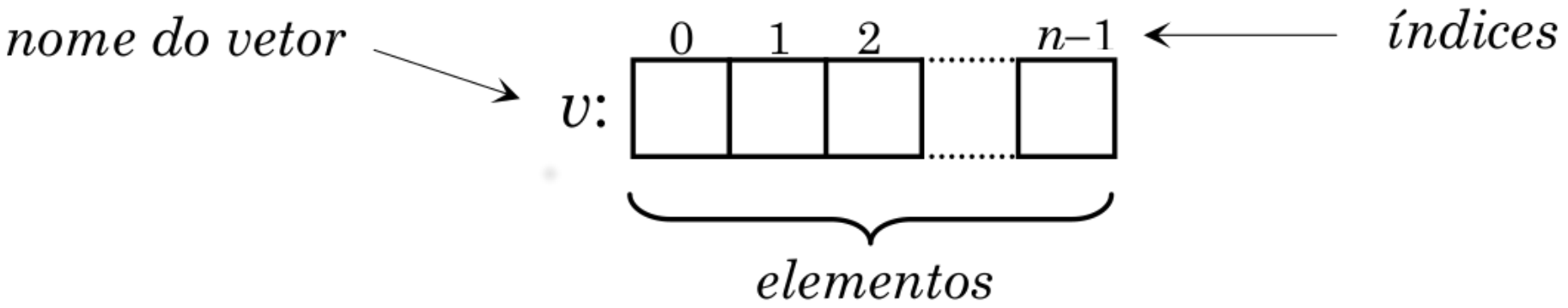
UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS

Aula - String

Definição de Vetor

- Variável composta homogênea unidimensional

"Coleção de variáveis de mesmo tipo, onde cada elemento é acessado por um único índice"



Declaração de Vetores

- A forma geral de uma declaração de vetores é

```
<tipo> <nome>[<inteiro>];
```

- Cada elemento de um vetor ocupa posições consecutivas de memória

int = 4 bytes ⟷ `int v[10];` ⟷ *40 bytes de memória*

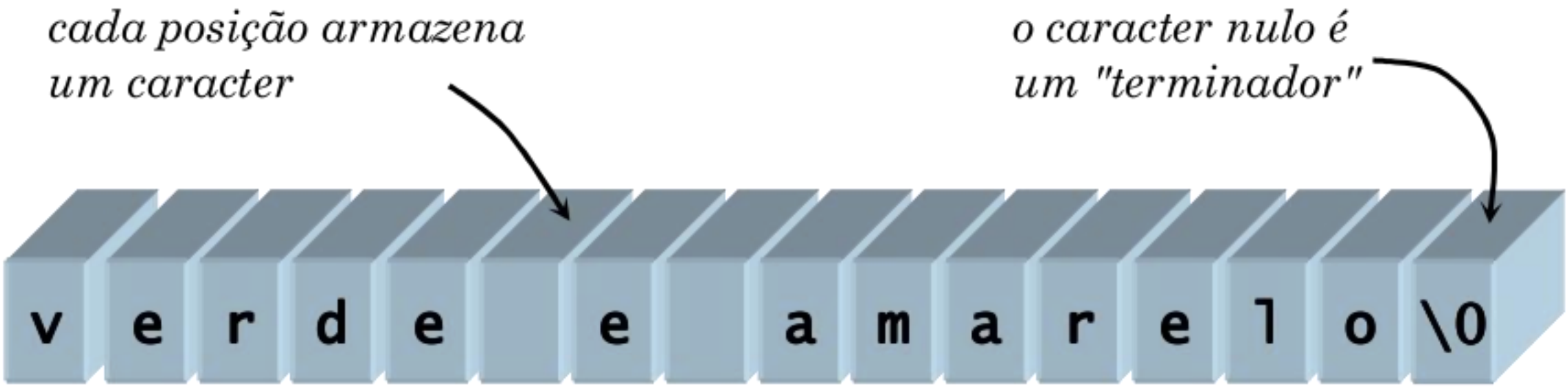
Aula passada... Inicialização de Vetores

```
<tipo> <nome>[<tamanho>] = {<elementos>};

float moeda[10] = {1.00, 0.50, 0.25, 0.10, 0.05};


<tipo> <nome>[] = {<elementos>};

char diaSemana[] = {'D', 'S', 'T', 'Q', 'Q', 'S', 'S'};
```

String

String

- Vetor de caracteres terminada pelo caractere nulo



String

- Vetor de caracteres terminada pelo caractere nulo

```
int main(){
    char nome1[10] = “Valeria";
    char nome2[10] = {‘V',’a',’l',’e',’r',’i’,’a'};
    char nome3[5]  = "Valeria Pinheiro";
return 0;
}
```

String

- Vetor de caracteres terminada pelo caractere nulo

```c
#include <stdio.h>

void escreve(char string[]){
    int i = 0;
    while(string[i] != '\0') putchar(string[i++]);
}

int main(){
    char nome1[10] = “Valeria";
    char nome2[10] = {‘V',’a',’l',’e',’r',’i’,’a'};
    char nome3[5]  = "Valeria Pinheiro";
    escreve(nome1);
    escreve(nome2);
    escreve(nome3);
    return 0;
}
```

Entrada / Saída

```c
#include <stdio.h>

#define SIZE 30

int main(){
    char nome[SIZE];

    printf("Digite seu nome: ");
    scanf("%s",nome);
    printf("Bom dia, %s.\n",nome);

    return 0;
}
```

Entrada / Saída

```c
#include <stdio.h>

#define SIZE 30

int main(){
    char nome[SIZE];

    printf("Digite seu nome: ");
    scanf("%s",nome);
    printf("Bom dia, %s.\n",nome);

    return 0;
}
```

Entrada / Saída

```c
#include <stdio.h>

#define SIZE 30

int main(){
    char nome[SIZE];

    printf("Digite seu nome: ");
    scanf("%s",nome);
    printf("Bom dia, %s.\n",nome);

    return 0;
}
```

Entrada / Saída

```c
#include <stdio.h>

#define SIZE 30

int main(){
    char nome[SIZE];

    printf("Digite seu nome: ");
    gets(nome);
    printf("Bom dia, %s.\n",nome);

    return 0;
}
```

Biblioteca <string.h>

memchr
memcmp
memcpy
memmove
memset
strcat
strchr
strcoll
strcmp
strcpy
strcspn
strerror
strlen
strncat
strncmp
strncpy
strpbrk
strrchr
strspn
strstr
strtok
strxfrm

Biblioteca <string.h>

- **Comprimento** de uma String

```
str = string
len = length (comprimento)
```

```
size_t strlen( const char str[] );
```

```
char nome1[30] = "Dragon Ball";
int  tam;

tam = strlen(nome1);

//tam = 11
```

- **Comprimento** de uma String

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

int main() {
    char str[] = "Olá, mundo!";
    int comprimento = strlen(str);

    printf("Comprimento da string: %d\n",
comprimento);

    return 0;
}
```

Biblioteca <string.h>

- **Comparação** entre Strings

```
str = string
cmp = comparação
```

```
int strcmp( const char str1[], const char str2[] );
```

valor	Significado
valor < 0	O primeiro caractere que não é igual tem um valor inferior em **string1** do que em **string2**
valor = 0	Os conteúdos de ambas as strings são iguais
valor > 0	O primeiro caractere que não é igual tem um valor superior em **string1** do que em **string2**

Biblioteca <string.h>

- **Comparação** entre Strings

```
'a' - 'c' = -2
'c' - 'a' = 2
```

```
int strcmp( const char str1[], const char str2[] );
```

valor	Significado
valor < 0	O primeiro caractere que não é igual tem um valor inferior em **string1** do que em **string2**
valor = 0	Os conteúdos de ambas as strings são iguais
valor > 0	O primeiro caractere que não é igual tem um valor superior em **string1** do que em **string2**

- **Comparação** entre Strings

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

int main() {
    char str1[] = "maçã";
    char str2[] = "banana";

    int resultado = strcmp(str1, str2);

    if (resultado < 0) {
        printf("str1 vem antes de str2\n");
    } else if (resultado > 0) {
        printf("str2 vem antes de str1\n");
    } else {
        printf("str1 e str2 são iguais\n");
    }

    return 0;
}
```

Biblioteca <string.h>

- **Cópia** de uma string para outra

str = string
cpy = copy (cópia)

```
char* strcpy( char str1[] , const char str2[] );
```

```
char nome1[30] = "Antonio
Carlos";
char nome2[30] = "Bruno";

strcpy(nome1,nome2);

//nome1 = Bruno\0o Carlos\0
```

- **Cópia** de uma string para outra

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

int main() {
    char origem[] = "Ola, mundo!";
    char destino[20];

    strcpy(destino, origem);

    printf("Origem: %s\n", origem);
    printf("Destino: %s\n", destino);

    return 0;
}
```

Biblioteca <string.h>

- Concatena uma string no final da outra

```
str = string
cat = concatenação
```

```
char* strcat( char str1[] , const char str2[] );
```

```
char nome1[30] = "Dragon ";
char nome2[30] = "Ball";

strcat(nome1,nome2);

//nome1 = "Dragon Ball"
```

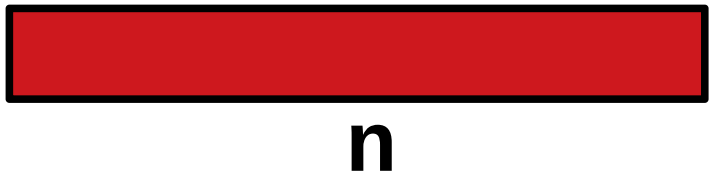
- **Concatena** uma string no final da outra

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

int main() {
    char str1[50] = "Olá, ";
    char str2[] = "mundo!";

    strcat(str1, str2);

    printf("String concatenada: %s\n", str1);

    return 0;
}
```

Biblioteca <string.h>

- Versões com número de caracteres delimitado

```
int strncmp( const char str1[], const char str2[], size_t n );

char* strncpy( char str1[], const char str2[], size_t n );

char* strncat( char str1[], const char str2[], size_t n );
```

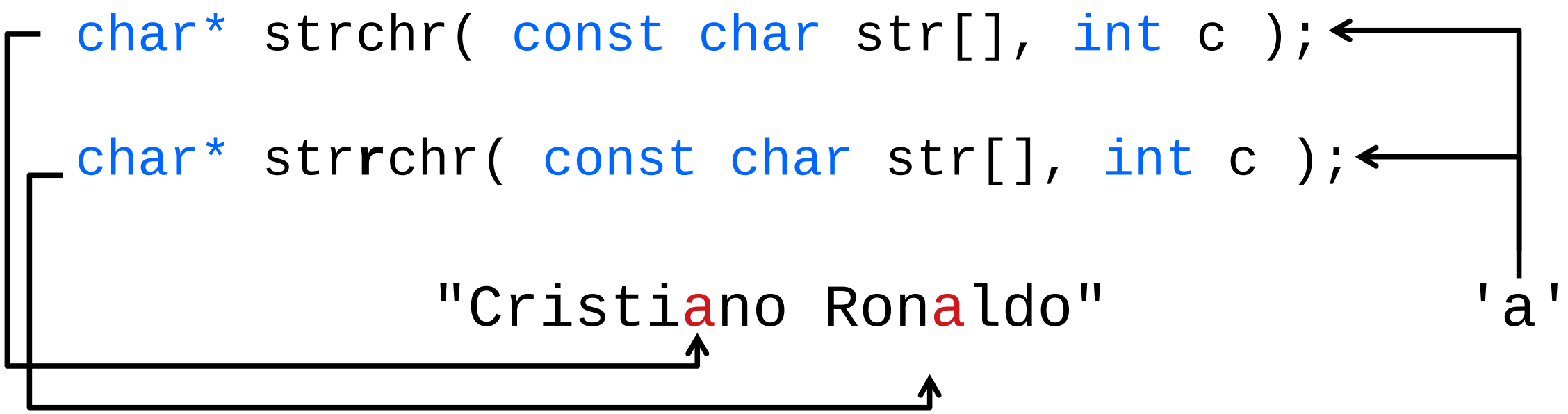


Biblioteca <string.h>

- Busca por um caractere

```
str = string
chr = character (caractere)
```

```
char* strchr( const char str[], int c );

char* strrchr( const char str[], int c );
```



- **Busca** por um caractere

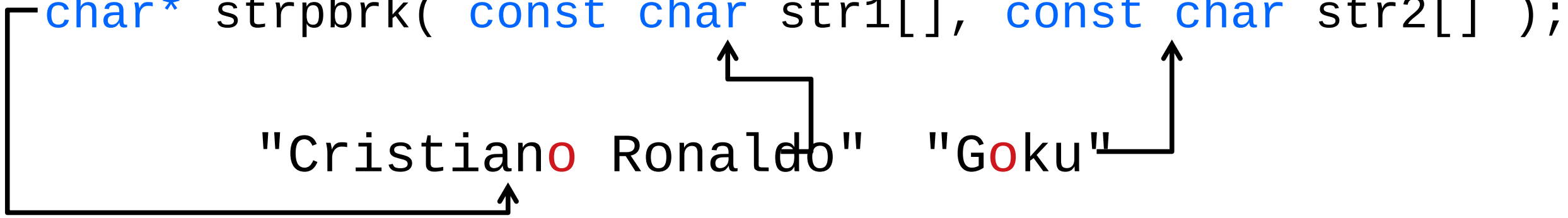
```c
#include <stdio.h>
#include <string.h>

int main() {
    char str[] = "Hello, world!";
    char caractere = 'o';

    char *resultado = strrchr(str, caractere);

    if (resultado != NULL) {
        printf("A última ocorrência de '%c' em '%s' está na posição %ld\n",
caractere, str, resultado - str);
    } else {
        printf("'%c' não foi encontrado em '%s'\n", caractere, str);
    }

    return 0;
}
```

Biblioteca <string.h>

- Busca por um caractere

str = string
pbrk = pointer break

```
char* strpbrk( const char str1[], const char str2[] );
```



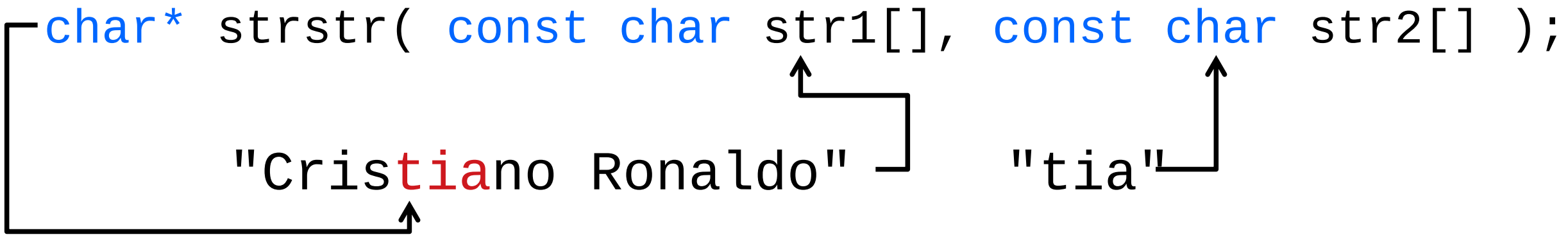
- **Busca** por um caractere

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

int main() {
    char str[] = "Ola, mundo!";
    char conjunto[] = "aeiou";

    char *resultado = strpbrk(str, conjunto);

    if (resultado != NULL) {
        printf("O primeiro caractere encontrado em '%s' que pertence ao conjunto '%s' é '%c'\n", str, conjunto, *resultado);
    } else {
        printf("Nenhum caractere do conjunto foi encontrado em '%s'\n", str);
    }

    return 0;
}
```

Biblioteca <string.h>

- Busca por uma substring

```
str = string
str = string (string na string)
```



- **Busca** por uma substring

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

int main() {
    char str[] = "Hello, world!";
    char subtring[] = "world";

    char *resultado = strstr(str, subtring);

    if (resultado != NULL) {
        printf("A subtring '%s' foi encontrada em '%s' na posição %ld\n",
subtring, str, resultado - str);
    } else {
        printf("A subtring '%s' não foi encontrada em '%s'\n", subtring, str);
    }

    return 0;
}
```

Biblioteca <string.h>

- Divide uma string

str = string
tok = token (símbolo)

```
char* strtok( char str1[], const char str2[] );
```

```
char str1[80] = "Nome - Matricula -
Nota";
char str2[]   = " - ";
char *t;

t = strtok(str1,str2);

while(t != NULL){
    printf("%s\n",t);
    t = strtok(NULL,str2);
}
```

Saída

```
Nome
Matricula
Nota
```

Exercícios

1 - Receba uma string e verifque se é maiúscula (se todos os seus caracteres são maiúculos). Em caso afirmativo, retorne 1. Em caso negativo, retorne 0.

2 - Escreva um programa que leia uma string e, em seguida, imprima a inversa da string lida.
Exemplo: Tangamandapio → oipadnamagnaT

3 - Escreva um programa que leia uma string e, em seguida, informe se é ou não um palíndromo.
Exemplo: arara → arara (SIM)

Exemplo: teste → etset (NÃO)

4 - Teste os exemplos dos slides com a biblioteca string.h:

`strlen, strcmp, strcat`

`strchr, strrchr e strpbrk`