

Strings

Na programação em linguagem C, uma string é uma sequência de caracteres terminada com um caractere nulo `\0`. Por exemplo:

```
char str[20] = "Linguagem C";
```

Quando o compilador encontra uma sequência de caracteres entre aspas duplas, ele acrescenta um caractere nulo `\0` no final por padrão.

0	1	2	3	4	5	6	7	8	9	10	11
L	i	n	g	u	a	g	e	m		C	\0

1 - Declaração de strings

O exemplo a seguir declara um string com capacidade para 10 caracteres.

```
char str[10];
```

2 - Inicializar strings

É possível inicializar strings de várias maneiras.

```
char str[5] = "abcd";

char str[] = "abcd";

char str[5] = {'a', 'b', 'c', 'd', '\0'}

char str[] = {'a', 'b', 'c', 'd', '\0'}
```

Assim como vetores, as strings não suportam o operador de atribuição depois que ela é declarada.

```
char str[5];
str = "abcd" /* Erro */
```

3 - Ler uma string do usuário

É possível usar a função `scanf()` com a especificação `%s` para ler uma string do usuário. No entanto, a função `scanf()` fará a leitura até encontrar um espaço em branco. Repare que é utilizado apenas `nome` ao invés de `&nome`. Isso porque o `nome` no `scanf()` aponta para o endereço de memória do primeiro elemento da string apenas.

```
#include <stdio.h>

int main()
{
    char nome[21];

    printf("Informe seu nome: ");
    scanf("%s", nome);

    printf("Seu nome é %s.", nome);

    return 0;
}
```

Uma forma mais eficiente de ler uma string do usuário é apresentada a seguir. Aqui, a string é lida até que seja encontrada uma quebra de linha.

```
#include <stdio.h>

int main()
{
    char nome[21];

    printf("Informe seu nome: ");
    scanf(" %20[^\n]", nome);

    printf("Seu nome é %s.", nome);

    return 0;
}
```

4 - Funções de string

A linguagem C oferece a biblioteca padrão `string.h` para manipulação de strings. Algumas das funções presentes nessa biblioteca são apresentadas a seguir.

4.1 - Tamanho da string

A função `strlen` calcula o comprimento de uma determinada string. Observe que a função `strlen()` não conta o caractere nulo `\0` ao calcular o comprimento da string.

Sintaxe: `strlen(string)`

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

int main()
{
    char str[] = "Universidade Positivo";

    printf("Tamanho da string: %d caracteres.", strlen(str));

    return 0;
}
```

4.2 - Copiar strings

A função `strcpy` copia toda a string de origem (incluindo o caracter nulo) para uma string de destino.

Sintaxe: `strcpy(string destino, string origem)`

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

int main()
{
    char str1[31] = "Universidade Positivo";
    char str2[31];

    strcpy(str2, str1);

    printf("%s", str2);

    return 0;
}
```

4.3 - Concatenar strings

A função `strcat` concatena uma cópia da segunda string ao final da primeira string.

Sintaxe: `strcat(str destino, str origem)`

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

int main()
{
    char str1[20] = "Universidade";
    char str2[30] = " Positivo";

    strcat(str1, str2);
```

```
printf("%s", str1);

return 0;
}
```

4.4 - Comparar strings

A função `strcmp()` compara duas strings, caractere por caractere, e retorna um valor. Caso o valor de retorno seja 0, as strings são iguais. Caso o valor de retorno seja menor que 0, a string `str1` é alfabeticamente menor que a string `str2`. Caso o valor de retorno seja maior que 0, a string `str1` é alfabeticamente maior que a string `str2`.

Sintaxe: `strcmp(str1, str2)`

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

int main()
{
    char str[30] = "Universidade Positivo";

    if(strcmp(str, "Universidade Positivo") == 0){
        printf("As strings são iguais.");
    }
    else{
        printf("As strings são diferentes.");
    }

    return 0;
}
```

5 - Tabela ASCII

ASCII foi o primeiro conjunto de caracteres (padrão de codificação) usado entre computadores na Internet e significa "American Standard Code for Information Interchange".

Foi projetado no início dos anos 60, como um conjunto de caracteres padrão para computadores e dispositivos eletrônicos e contém os números de 0 a 9, as letras maiúsculas e minúsculas do inglês de A a Z e alguns caracteres especiais.

Coluna 1	Coluna 2	Coluna 3	Coluna 4	Coluna 5
32 = space	33 = !	34 = "	35 = #	36 = \$
37 = %	38 = &	39 = '	40 = (41 =)
42 = *	43 = +	44 = ,	45 = -	46 = .
47 = /	48 = 0	49 = 1	50 = 2	51 = 3
52 = 4	53 = 5	54 = 6	55 = 7	56 = 8
57 = 9	58 = :	59 = ;	60 = <	61 = =
62 = >	63 = ?	64 = @	65 = A	66 = B
67 = C	68 = D	69 = E	70 = F	71 = G
72 = H	73 = I	74 = J	75 = K	76 = L
77 = M	78 = N	79 = O	80 = P	81 = Q
82 = R	83 = S	84 = T	85 = U	86 = V
87 = W	88 = X	89 = Y	90 = Z	91 = [
92 = \	93 =]	94 = ^	95 = _	96 = `
97 = a	98 = b	99 = c	100 = d	101 = e
102 = f	103 = g	104 = h	105 = i	106 = j
107 = k	108 = l	109 = m	110 = n	111 = o
112 = p	113 = q	114 = r	115 = s	116 = t
117 = u	118 = v	119 = w	120 = x	121 = y
122 = z	123 = {	124 =	125 = }	126 = ~
127 = del				

O código a seguir gera a tabela ASCII em linguagem C.

```
#include <stdio.h>

int main() {

    int i, count;

    printf("32 = space\t");

    count = 1;
```

```

for(i = 33; i < 127; i++){
    printf("%d = %c\t\t", i, i);
    count++;

    if(count == 5){
        printf("\n");
        count = 0;
    }
}

printf("127 = del");

return(0);
}

```

6 - Atividades

1. Escreva um programa que mostre o tamanho de uma string informada pelo usuário (utilize a função apropriada).
2. Escreva um programa que mostre o tamanho de uma string informada pelo usuário (não utilize funções).
3. Escreva um programa para exibir os caracteres de A a Z.
4. Escreva um programa que receba a string = "Hello, World!" e a apresente invertida.
5. Escreva um programa que verifique se duas strings são iguais.
6. Escreva um programa que leia uma frase de até 50 caracteres. Em seguida, escreva a frase sem os espaços em branco.
7. Escreva um programa que receba uma string e um caractere, e retorne o número de vezes que esse caractere aparece na string.
8. Escreva um programa para contar a quantidade de vogais em uma string.
9. Um hospital local está fazendo uma campanha para receber doação de sangue. O propenso doador deve inicialmente se cadastrar informando o seu primeiro nome, sua idade, seu peso, responder a um breve questionário e apresentar um documento oficial com foto. Faça um programa em C que permita ao hospital avaliar a aptidão de um voluntário quanto à doação de sangue. Para estar apto a doar sangue, o voluntário deve ter idade entre 16 e 69 anos, pesar pelo menos 50 kg, estar bem alimentado e não estar resfriado. O programa deve ler os dados e imprimir no final o nome do voluntário e se ele está apto ou não.
10. Escreva um programa que, dada uma string, diga se ela é um palíndromo ou não. Lembrando que um palíndromo é um texto que tenha a propriedade de poder ser lido tanto da direita para a esquerda como da esquerda para a direita. **Exemplos:** ovo, arara, anotaram a data da maratona.