

Fundamentos de Programação

Linguagem de Programação C
String

Sumário

1 Definição

2 Strings

3 Manipulação

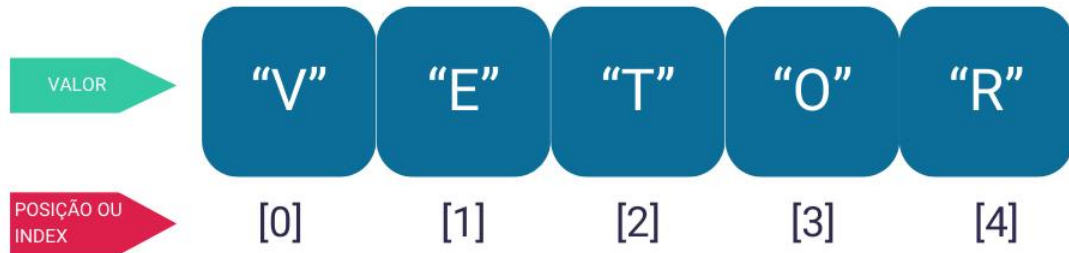
4 Biblioteca

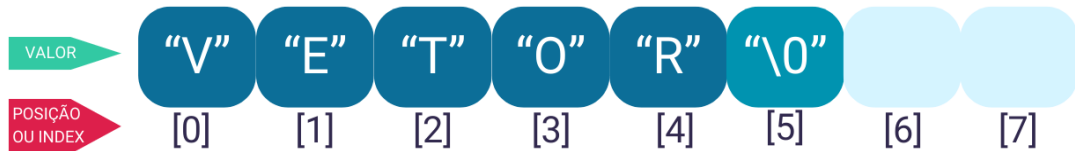
5 Desafio

Definição

O que é uma String?

- › Um vetor de caracteres é uma string quando armazenamos uma **sequência de caracteres** em posições contíguas na memória, terminada com o caractere nulo `\0` para indicar o final da sequência.





Strings

1

```
char palavra[] =  
    {'P', 'r', 'o', 'g', 'r', 'a', 'm', 'a', 'r', '\0'};
```

Código 1: Exemplo de declaração como um vetor.

Nota:

Essa forma de declaração não é prática.

```
1 char palavra[] = "Programar";
```

Código 2: Exemplo de declaração como string literal.

```
1 #include <stdio.h>
2 int main(){
3     char nome_animal[10];
4     printf("Digite o nome de um animal:\n");
5     scanf("%s", nome_animal);
6     return 0;
7 }
```

Código 3: Exemplo de entrada de string usando %s.

Implementação trabalhosa sem usar %s.

```
1 #include <stdio.h>
2 int main(){
3     char nome_animal[] = "Gato";
4     int i = 0;
5     while (nome_animal[i] != '\0')
6         printf("%c", nome_animal[i++]);
7     return 0;
8 }
```

Código 4: Exemplo de saída elementos individuais usando \0.

Implementação adequada.

```
1 #include <stdio.h>
2 int main()
3 {
4     char nome_animal[10];
5     printf("Digite o nome de um animal:\n");
6     scanf("%s", nome_animal);
7     printf("%s\n", nome_animal);
8     return 0;
9 }
```

Código 5: Exemplo de saída de string usando %s.

Manipulação

```
1 #include <stdio.h>
2 int contar(char *palavra){
3     int i = 0;
4     while(palavra[i] != '\0'){
5         i++;
6     }
7     return i;
8 }
9 int main(){
10     char palavra[100];
11     printf("Digite uma palavra:\n");
12     scanf("%s", palavra);
13     printf("Total de caracteres: %d\n", contar(palavra));
14     return 0;
15 }
```

Código 6: Exemplo de contagem de caracteres.

```
1 #include <stdio.h>
2 int comparar(char *palavra_a, char *palavra_b){
3     int i = 0;
4     while(palavra_a[i] != '\0' || palavra_b[i] != '\0'){
5         if(palavra_a[i] != palavra_b[i])
6             return 0;
7         i++;
8     }
9     return 1;
10 }
11 int main(){
12     char palavra_a [100], palavra_b[100];
13     scanf("%s %s", palavra_a, palavra_b);
14     if (comparar(palavra_a,palavra_b)){
15         printf("Eh igual!\n");
16         return 0;
17     }
18     printf("Nao eh igual!\n");
19     return 0;
20 }
```

Código 7: Exemplo de comparação.

```
1 #include <stdio.h>
2 void copiar(char *origem, char *destino){
3     int i = 0;
4     while(origem[i] != '\0'){
5         destino[i] = origem[i];
6         i++;
7     }
8     destino[i] = '\0';
9 }
10 int main(){
11     char origem [10], destino[10];
12     scanf("%s %s", origem, destino);
13     copiar(origem , destino);
14     return 0;
15 }
```

Código 8: Exemplo de cópia.

Biblioteca

- › Existem funções que facilitam a manipulação de strings.
- › A biblioteca `string.h` fornece funções para contar caracteres, copiar, concatenar strings, entre outras operações.

```
1 #include <stdio.h>
2 #include<string.h>
3 int main(){
4     char palavra[100];
5     printf("Digite uma palavra:\n");
6     scanf("%s", palavra);
7     int quant_caracteres = strlen(palavra);
8     printf("Total de caracteres: %d\n", quant_caracteres);
9     return 0;
10 }
```

Código 9: Exemplo de uso de strlen().

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char palavra_a [10], palavra_b[10];
5     scanf("%s %s", palavra_a, palavra_b);
6     if (strcmp(palavra_a,palavra_b) != 0){
7         printf("Nao eh igual!\n");
8     }else{
9         printf("Eh igual!\n");
10    }
11    return 0;
12 }
```

Código 10: Exemplo de uso strcmp().

```
1 #include <stdio.h>
2 #include <string.h>
3 int main(){
4     char origem [10], destino[10];
5     scanf("%s %s", origem, destino);
6     strcpy(destino,origem);
7     return 0;
8 }
```

Código 11: Exemplo de uso strcpy().

```
1 #include <stdio.h>
2 #include <string.h>
3 int main() {
4     char origem[50] = "Olá, ";
5     char destino[50] = "mundo!";
6     strcat(origem, destino);
7     printf("%s\n", destino);
8     return 0;
9 }
```

Código 12: Exemplo de uso strcat().

Função	Descrição
<code>fgets(str, n, stdin)</code>	Lê até $n-1$ caracteres da entrada padrão e armazena na string <code>str</code> .
<code>puts(str)</code>	Imprime a string <code>str</code> na saída padrão, seguida por uma nova linha.
<code>scanf("%s", str)</code>	Lê uma palavra (até o primeiro espaço em branco) da entrada padrão e armazena na string <code>str</code> .

Tabela 1: Funções de Entrada e Saída para Manipulação de Strings.

```
1 #include <stdio.h>
2 int main(int argc, char const *argv[])
3 {
4     char str[20];
5     //Entrada padrão com espaços.
6     fgets(str, sizeof(str), stdin);
7     printf("%s\n", str);
8     return 0;
9 }
```

Código 13: Exemplo de uso fgets().

```
1 #include <stdio.h>
2 int main(int argc, char const *argv[])
3 {
4     char c;
5     printf("Digite um caractere:\n");
6     scanf(" %c", &c);
7     // fflush(stdin); // Limpar buffer no Windows - MinGW.
8     // __fpurge(stdin); // Limpar buffer no Linux - Depende do Compilador.
9     //getc(stdin) // Alternativa portátil
10    getchar(); // Alternativa portátil
11    printf("Digite um caractere:\n");
12    scanf(" %c", &c);
13    return 0;
14 }
```

Código 14: Exemplo de uso fgets().

```
1 #include <stdio.h>
2 #define TOTAL_NOMES 5
3 #define TAMANHO_MAXIMO 20
4 int main(int argc, char const *argv[])
5 {
6     char nomes[TOTAL_NOMES][TAMANHO_MAXIMO];
7     printf("-----INSERIR-----\n");
8     for (int i = 0; i < TOTAL_NOMES; i++)
9     {
10         printf("Nome %d ?", i + 1);
11         scanf("%s", nomes[i]);
12     }
13     printf("-----LISTAR-----\n");
14     for (int i = 0; i < TOTAL_NOMES; i++)
15         printf("%s\n", nomes[i]);
16     return 0;
17 }
```

Código 15: Exemplo para preencher e mostrar lista de strings.

Desafio

- Crie um programa que recebe como entrada uma palavra e, em seguida verifica se essa palavra é **palíndromo**.
- A verificação deve ser feita em uma função.

(Deitel; Deitel, 2011) - Capítulo 8



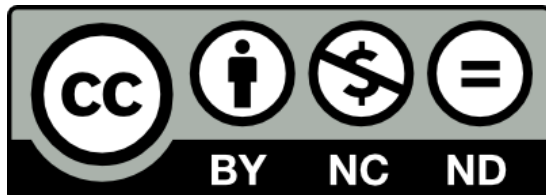


DE OLIVEIRA, J.F.; MANZANO, J.A.N.G. **Algoritmos: Lógica para Desenvolvimento de Programação de Computadores**. 16. ed. São Paulo: Editora Érica, 2004.



DEITEL, Paul; DEITEL, Harvey. **C: Como Programar**. 6. ed. São Paulo: Pearson Universidades, 2011.

Estes slides estão protegidos por uma licença Creative Commons



Este modelo foi adaptado de Maxime Chupin.

Fundamentos de Programação

Linguagem de Programação C
String