

Linguagem e Técnicas de Programação



Manipulação de Strings

Uma string em linguagem C é uma sequência de caracteres terminada por um caractere nulo ('\0'). Elas são importantes na programação porque são usadas para representar texto e dados de maneira legível. Strings são usadas para entrada/saída, formatação de texto, processamento de informações, entre outros.

Manipulação de Strings em C

Strings como Arrays de Caracteres:


Em linguagem C, uma string é representada como um array (ou seja, uma sequência) de caracteres. Cada caractere é armazenado em uma posição (índice) do array. Por exemplo, a string "Hello" é armazenada como o array de caracteres ['H', 'e', 'l', 'l', 'o'].

Exemplo de declaração e inicialização de uma string em C:

```
1  #include <stdio.h>
2
3  int main() {
4      char mensagem[] = "Hello";
5
6      printf(mensagem);
7  }
```

Manipulação de Strings em C

Observe que o tamanho do array é automaticamente calculado pelo compilador com base no tamanho da string e um espaço extra é reservado para o terminador nulo.



Declarando Arrays de Caracteres:

Para declarar uma string em C, você declara um array de caracteres. Um array é uma coleção de elementos do **mesmo tipo**, e cada caractere da string é armazenado como um elemento desse array.

```
1 #include <stdio.h>
2
3 int main() {
4
5     // Declaração de um array
6     //de caracteres para armazenar uma string
7     char mensagem[20];
8
9 }
```

Manipulação de Strings em C

Nesse exemplo, myString é um array de caracteres com espaço para até 20 caracteres.



Inicializando com Literais de String:

Você pode inicializar uma string diretamente com um literal de string, que é uma sequência de caracteres delimitada por **aspas duplas**.

```
1 #include <stdio.h>
2
3 int main() {
4
5     char mensagem[] = "Hello, world!";
6
7 }
```

Manipulação de Strings em C

Nesse caso, o compilador determinará automaticamente o tamanho necessário para o array de caracteres com base no comprimento da string literal, incluindo o terminador nulo.

Acesso aos Caracteres de uma String

Dado o array de caracteres

char palavra[] = "Código";

acesse o segundo caractere e exiba-o.

```
1  #include <stdio.h>
2
3  int main() {
4      char palavra[] = "Código";
5
6      char segundo_caractere = palavra[1];
7      printf("Segundo caractere: %c\n", segundo_caractere);
8
9      return 0;
10 }
```

Manipulação de Strings em C

Exemplo de saída com laço de repetição

Crie um programa que imprima na tela, a palavra recebida através de uma variável. Uma letra por linha e também na mesma linha.

```
3  
4 int main(){  
5     setlocale(LC_ALL, "Portuguese");  
6  
7     char palavra[] = "Código";  
8     int cont;  
9  
10    printf("A palavra contém: %i letras.\n", strlen(palavra));  
11  
12    for(cont=0; cont<strlen(palavra); cont++){  
13        .....  
14        printf("Letra %i: %c\n", cont, palavra[cont]);  
15    }  
16  
17  
18 }
```

Função:

strlen – conta quantidade de caracteres de um vetor.

Necessário incluir a biblioteca string.h

Concatenação de Strings

Crie um programa que solicite ao usuário seu primeiro nome e sobrenome separadamente, e depois concatene-os para formar o nome completo.

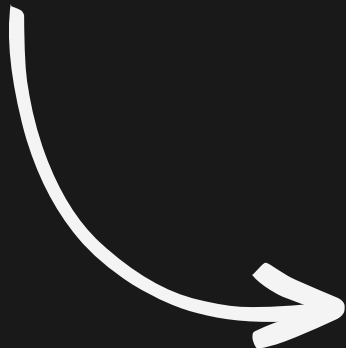
```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char primeiro_nome[50];
6      char sobrenome[50];
7      char nome_completo[100];
8
9      printf("Digite seu primeiro nome: ");
10     scanf("%s", primeiro_nome);
11
12     printf("Digite seu sobrenome: ");
13     scanf("%s", sobrenome);
14
15     strcpy(nome_completo, primeiro_nome);
16     strcat(nome_completo, " ");
17     strcat(nome_completo, sobrenome);
18
19     printf("Nome completo: %s\n", nome_completo);
20
21     return 0;
22 }
```

Função:
strcpy – “copia” um texto de uma origem ou variável.

Função:
strcat – concatena um valor (adiciona) a uma variável existente.

Crie um programa que peça ao usuário uma palavra e conte quantas vogais ela contém.

Tamanho e Iteração de Strings



```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5      char palavra[100];
6      printf("Digite uma palavra: ");
7      scanf("%s", palavra);
8
9      int contador_vogais = 0;
10     int comprimento = strlen(palavra);
11
12     for (int i = 0; i < comprimento; i++) {
13         char caractere = palavra[i];
14         if (caractere == 'a' || caractere == 'e' || caractere == 'i' || caractere == 'o' || caractere == 'u' ||
15             caractere == 'A' || caractere == 'E' || caractere == 'I' || caractere == 'O' || caractere == 'U') {
16             contador_vogais++;
17         }
18     }
19
20     printf("Número de vogais: %d\n", contador_vogais);
21     return 0;
22 }
```