

Tipos de Dados e Manipulação de Strings

Prof. MSC. Wagner Guimarães Al-Alam

Universidade Federal do Ceará
Campus de Quixadá

20 de maio de 2016



Tipos de dados

Inteiro	int	(-32.768 a 32.767)
Ponto Flutuante	float	(3.4 E-38 a 3.4E+38)
Precisão Simples	double	(1.7 E-308 a 1.7E+308)
Precisão Dupla	char	(-128 a 127)
Caractere ASCII	boolean	(true ou false)



Manipulação de Texto em C

- Em C, não existe um tipo de dados específico para texto.
 - Para contornar esta deficiência, utilizamos uma biblioteca que facilita a vida do programador.
 - String.h (`#include <string.h>`)
 - Vetor de char terminado por null e interpretado como texto



Declarando uma String

- Como a string utiliza um vetor de char para declarar uma variável que será utilizada para armazenar texto, declaramos uma variável deste tipo.
- Formato da declaração da variável:
char nome_da_variável[TAMANHO]
- Deve-se atentar ao fato do tamanho ser maior que a quantidade máxima de caracteres esperados na string.



Leitura/Escreita de uma String com **scanf()**

- Para ler uma string e armazenar em um vetor de char utilizamos o seguinte comando:
scanf(“%s”, &nome_da_variavel);
- Para escrever uma string na tela, utilizamos o seguinte comando:
printf(“%s”, nome_da_variavel);



Leitura/Escreita de uma String com **fgets()** e **puts()**

- Para ler uma string e armazenar em um vetor de char utilizamos o seguinte comando:
fgets(nome_da_variavel, tamanho, stdin);
- Para escrever uma string na tela, utilizamos o seguinte comando:
puts(nome_da_variavel);




Funções da Biblioteca String.h

- Cópia de valores de uma string para outra
 - strcpy(s1, s2)
 - Copia s2 em s1
- Concatenação de uma string ao final de outra
 - strcat(s1, s2)
 - Concatena s2 ao final de s1
- Descobrir o tamanho de uma string
 - strlen(s1)
 - Retorna o tamanho de s1
- Comparação de strings
 - strcmp(s1, s2)
 - Retorna 0 se s1 e s2 são iguais; menor que 0 se s1 < s2; maior que 0 se s1 > s2
 - strcmp() retorna falso se as strings são iguais, então, assegure-se de usar o operador ! Para negar o resultado e reverter essa condição.



Exemplo

```
Main.c 
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(void){
5      char s1[80], s2[80];
6      scanf("%s", &s1);
7      scanf("%s", &s2);
8      printf("comprimentos: %d %d\n", strlen(s1), strlen(s2));
9      if(!strcmp(s1, s2)){
10         printf("As strings são iguais\n");
11     }
12     strcat(s1, s2);
13     printf("%s\n", s1);
14     strcpy(s1, "Isso é um teste. \n");
15     printf(s1);
16     return 0;
17 }
```



Acessando um Caractere da String

- Para acessar um único caractere de uma string, utilizamos a notação de vetores, passando a posição do caractere que se deseja acessar, considerando a posição inicial como 0;
- `char minha_string[100];`
 - Para acessar o primeiro caractere utiliza-se o seguinte comando:
`minha_string[0];`
 - Para acessar o segundo caractere utiliza-se o seguinte comando:
`minha_string[1];`
 - Segue até o último elemento (tamanho - 1)



Exercícios:

- Faça um programa que solicite ao usuário digitar os seguintes dados:
 - Primeiro nome
 - Nome do meio
 - Último sobrenome
 - Rua em que reside
 - Número da casa
 - CEP
- Ao final, seu programa deve imprimir a seguinte frase, substituindo os * pelos campos acima na mesma ordem:
 - O aluno * * * reside na rua *, * cujo CEP é *.

