Algoritmos e Programação

Prof. Eduardo Magalhães

Linguagem C

- Instruções de entrada e saída

Aula 03

Instruções de Entrada e Saída

- Caracteres (char)

  Os caracteres são um tipo de dado: o **char**. O C trata os caracteres ('a', 'b', 'x', etc ...) como sendo variáveis de um *byte* (8 *bit*s).

  Os inteiros (**int**s) têm um número maior de *bytes*.

Instruções de Entrada e Saída

- Caracteres (char)

Além disso podemos usar um **char** para armazenar valores numéricos inteiros, além de usá-lo para armazenar caracteres de texto.

Exemplo 1:

```
#include <stdio.h>
int main ()
{
   char Ch;
   Ch='D';
   printf ("%c",Ch);

   return(0);
}
```

O que ele retorna?

Exemplo 2:

```
#include <stdio.h>
int main ()
{
   char Ch;
   Ch='D';
   printf ("%d",Ch);

   return(0);
}
```

O que ele retorna?

Instruções de Entrada e Saída

- Usando funções getch() e getche()

  Às vezes precisamos ler um caractere fornecido pelo usuário, para isso as funções mais usadas são getch() e getche(). Ambas retornam o caractere pressionado.

Instruções de Entrada e Saída

- **getch()** apenas retorna o caractere pressionado sem imprimí-lo na tela

- **getche()** imprime o caractere na tela antes de retorná-lo

Estas as funções podem ser encontradas no arquivo de cabeçalho **conio.h**

Exemplo

```
#include <stdio.h>
#include <conio.h>
int main ()
{
   char Ch;
   Ch=getch();
   printf ("Voce pressionou a tecla %c",Ch);
   return(0);
}
```

Instruções de Entrada e Saída

- String

No C uma string é um vetor de caracteres terminado com um caractere nulo.

O caracter nulo é um caractere com valor inteiro igual a zero (código ASCII igual a 0).

Instruções de Entrada e Saída

- String

  Para declarar uma string, podemos usar o seguinte formato geral:

  char nome_da_string[tamanho];

Instruções de Entrada e Saída

- String

  Isto declara um vetor de caracteres (uma string) com número de posições igual a *tamanho*.

  Temos que reservar um caractere para ser o terminador nulo **/0** (barra zero).

Instruções de Entrada e Saída

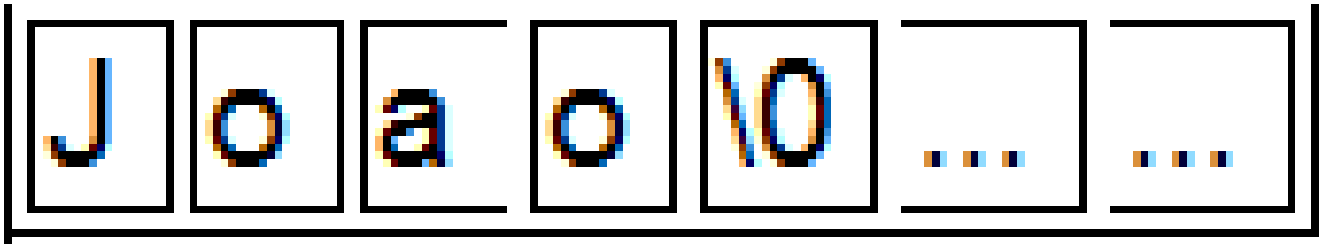
- String

  Temos que declarar o comprimento da string como sendo, no mínimo, um caractere maior que a maior string que pretendemos armazenar.

Instruções de Entrada e Saída

- String

Vamos supor que declaremos uma string de 7 posições e coloquemos a palavra João nela.



No caso acima, as duas células não usadas têm valores indeterminados. Isto acontece porque o C *não* inicializa variáveis.

Instruções de Entrada e Saída

- String

  Se quisermos ler uma string fornecida pelo usuário podemos usar a função **gets()**.

  A função **gets()** coloca o terminador nulo na string, quando você aperta a tecla "Enter".

  Não precisa do cabeçalho **conio.h**

Exemplo

```
#include <stdio.h>
int main ()
{
      char name[100];
      printf ("Digite uma palavra: ");
      gets (name);
      printf ("\n\nVoce digitou %s",name);
      return(0);
  }
```

Instruções de Entrada e Saída

- String

  No programa anterior, o tamanho máximo da string que você pode entrar é uma string de 99 caracteres.

  Como as strings são vetores de caracteres, para se acessar um determinado caracter de uma string, basta "indexarmos", ou seja, usarmos um índice para acessarmos o caracter desejado dentro da string

Exemplo

```
#include <stdio.h>
int main()
{
	char str[10] = "Joao";
	printf("\n\nString: %s", str);
	printf("\nSegunda letra: %c", str[1]);
	str[1] = 'U';
	printf("\nAgora a segunda letra eh: %c", str[1]);
	printf("\n\nString resultante: %s", str);
	return(0);
}
```

Instruções de Entrada e Saída

- Printf

  A função **printf()** tem a seguinte forma geral:

  printf (string_de_controle,lista_de_argumentos);

Instruções de Entrada e Saída

- Printf()

Teremos, na string de controle, uma descrição de tudo que a função vai colocar na tela. A string de controle mostra não apenas os caracteres que devem ser colocados na tela, mas também quais as variáveis e suas respectivas posições.

Instruções de Entrada e Saída

- Printf()

  Isto é feito usando-se os códigos de controle, que usam a notação **%**. Na string de controle indicamos quais, de qual tipo e em que posição estão as variáveis a serem apresentadas.

Instruções de Entrada e Saída

Apresentamos agora alguns dos códigos **%**:

Código	Significado
%d	Inteiro
%f	Float
%c	Caractere
%s	String
%%	Coloca na tela um %

Exemplos

printf ("Teste %% %%") -> "Teste % %“

printf ("%f",40.345) -> "40.345“

printf ("Um caractere %c e um inteiro %d",'D',120) ->
"Um caractere D e um inteiro 120“

printf ("%s e um exemplo","Este") -> "Este e um
exemplo“

printf ("%s%d%%","Juros de ",10) -> "Juros de 10%"

Instruções de Entrada e Saída

- Scanf()

  Usando a função **scanf()** podemos pedir dados ao usuário.

  Mais uma vez, devemos ficar atentos a fim de colocar o mesmo número de argumentos que o de códigos de controle na string de controle.

Instruções de Entrada e Saída

- Scanf()

  Outra coisa importante é lembrarmos de colocar o **&** antes das variáveis da lista de argumentos.

  O formato geral da função **scanf()** é:

  scanf (string-de-controle,lista-de-argumentos);

Exercícios propostos

- Escreva um programa que leia um caracter digitado pelo usuário, imprima o caracter digitado e o código ASCII correspondente a este caracter.

- Escreva um programa que leia duas strings e as coloque na tela. Imprima também a segunda letra de cada string.