

Fundamentos de Programação

Entrada e Saída de Dados

Dainf - UTFPR

Profa. Leyza Baldo Dorini
Prof. Bogdan Tomoyuki Nassu

Entrada de dados

Entrada de dados

Programas comerciais recebem entradas através da interação com o usuário ou lendo dados de arquivos.

Nesta disciplina, não consideraremos programas com interfaces gráficas. Arquivos serão usados mais adiante.

Apenas para termos alguma interação, trabalharemos com entrada simples de dados pela entrada padrão (teclado).


- Não trabalharemos com interfaces complexas em modo texto! É meio arcaico, e C não é exatamente a linguagem mais adequada para este tipo de tarefa...

Mesmo que a aparência dos nossos programas seja mais simples do que você pode estar acostumado, a lógica por trás deles é a mesma de qualquer programa comercial.

Entrada de dados

Na linguagem C, a função `scanf` é utilizada para leitura de dados:

a chamada da função `scanf` é composta por pelo menos dois argumentos, separados por vírgula...




```
1 scanf ( "%d", &var );
```

Entrada de dados

Na linguagem C, a função `scanf` é utilizada para leitura de dados:

```
1 scanf( "%d", &var );
```



endereço da variável que irá armazenar o valor lido. Para tal, o nome da variável deve ser precedido pelo símbolo `&`. Veremos isso melhor no futuro, mas por enquanto **não se esqueça do `&`**

Entrada de dados

Na linguagem C, a função `scanf` é utilizada para leitura de dados:

```
1 scanf( "%d", &var );
```

especificador de formato, o qual deve ser compatível com o tipo da variável que armazenará o valor lido:

%d para int
%f para float

Exemplo: função scanf

IMPORTANTE:


precisamos da incluir o arquivo `stdio.h`,
o qual possui definições de subrotinas
relativas a operações de entrada e saída

```
1  #include <stdio.h>
2  int main()
3  {
4      float peso;
5      int idade;
6      int cod;
7
8      scanf("%f", &peso);
9      scanf("%d", &idade);
10     scanf("%d", &cod);
11
12     return 0;
13 }
```

Exemplo: função scanf

```
1  #include <stdio.h>
2  int main()
3  {
4      float peso;
5      int idade;
6      int cod;
7
8      scanf("%f", &peso);
9      scanf("%d", &idade);
10     scanf("%d", &cod);
11
12     return 0;
13 }
```

leitura de um float, o qual
é armazenado na variável peso



leitura de um int



Não se esqueça do &


Um erro muito comum para quem está começando a programar em C é **esquecer o &!** Fique atento!



Lendo mais de um valor no mesmo scanf

Podemos ler mais de um valor no mesmo `scanf()`. Por exemplo, a leitura de dois inteiros e um float realizada no exemplo anterior poderia ser feita da seguinte forma:

```
1 scanf("%d %d %f", &cod, &idade, &peso);
```



endereços das variáveis, na ordem
em que a leitura será realizada.

Note que eles são separados por vírgula.
Não se esqueça do & (em cada variável)

Lendo mais de um valor no mesmo scanf

Podemos ler mais de um valor no mesmo `scanf()`. Por exemplo, a leitura de dois inteiros e um float realizada no exemplo anterior poderia ser feita da seguinte forma:

```
1 scanf("%d %d %f", &cod, &idade, &peso);
```

especificadores de formato.

Importante: eles devem ser compatíveis com o tipo das variáveis (na ordem em que elas aparecem listadas após a vírgula)

observe que eles estão separados por espaços

Lendo mais de um valor no mesmo scanf

Atenção

Ao colocar caracteres entre os especificadores, você obrigatoriamente terá que digitá-los junto com os valores.

```
1 scanf( "%d, %d, %f", &cod, &idade, &peso);
```

neste caso, ao digitar os valores no teclado, você deverá obrigatoriamente separá-los por vírgula (ou seja, vai ter que digitar 10, 25, 5.6)

Lendo mais de um valor no mesmo scanf

Essa característica pode ser útil em alguns casos. Por exemplo, leitura de datas:

```
1 scanf( "%d / %d / %d", &dia, &mes, &ano);
```

Ao digitar uma data - por exemplo, 23/10/2015 - o compilador armazenará os valores 23, 10 e 2015 nas respectivas variáveis, ignorando o símbolo /

Exemplo: função scanf

```
1  #include <stdio.h>
2  int main()
3  {
4      int idade, matricula;
5      float peso;
6      float salario;
7
8      scanf("%d", &matricula);
9      scanf("%d %f", &idade, &peso);
10     scanf("%f", &salario);
11
12     return 0;
13 }
```

precisamos da incluir o arquivo `stdio.h`,
o qual possui definições de subrotinas
relativas a operações de entrada e saída

leitura de um inteiro, o qual
é armazenado na variável `matricula`

leitura de um int e
de um float

leitura de um float

Exercícios de fixação

Exercício de fixação 1

Faça um programa que declare três variáveis do tipo float e três do tipo int. Depois, leia valores do teclado.

Um exemplo (você pode fazer outras variações):

```
1  #include<stdio.h>
2  int main(){
3
4      float peso, altura, taxa;
5      int idade, codigo;
6      int nroConta;
7
8      scanf("%f %f", &peso, &altura);
9      scanf("%f", &taxa);
10     scanf("%d", &idade);
11     scanf("%d", &codigo);
12     scanf("%d", &nroConta);
13
14     return 0;
15 }
```


Exercício de fixação 1

Faça um programa que declare três variáveis do tipo float e três do tipo int. Depois, leia valores do teclado.

Um exemplo (você pode fazer outras variações):

```
1  #include<stdio.h>
2  int main(){
3
4      float peso, altura, taxa;
5      int idade, codigo;
6      int nroConta;
7
8      scanf("%f %f", &peso, &altura);
9      scanf("%f", &taxa);
10     scanf("%d", &idade);
11     scanf("%d", &codigo);
12     scanf("%d", &nroConta);
13
14     return 0;
15 }
```

Exercício de fixação 2

Quais os erros no programa abaixo?

```
1
2 int main()
3 {
4     int idade, matricula;
5     float peso;
6     int cod;
7
8     scanf("%d", cod);
9     scanf("%d %f", &peso, &idade);
10    scanf("%d", &Matricula);
11
12    return 0;
13 }
```

Exercício de fixação 2

Quais os erros no programa abaixo?

```
1  #include <stdio.h>
2  int main()
3  {
4      int idade, matricula;
5      float peso;
6      int cod;
7
8      scanf("%d", cod);
9      scanf("%d %f", &peso, &idade);
10     scanf("%d", &Matricula);
11
12     return 0;
13 }
```

para usar o `scanf()`, precisamos incluir o arquivo `stdio.h`

faltou colocar o `&` antes do nome da variável

os especificadores não são compatíveis com os tipos das variáveis (estão na ordem trocada)

cuidado com o nome das variáveis (C é *case sensitive*)

Saída de dados

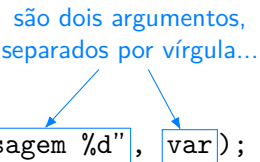
Anteriormente, estudamos o uso do `printf` para impressão de constantes numéricas e de resultados de expressões aritméticas. Agora, veremos como usar para imprimir o conteúdo de variáveis (ou de expressões aritméticas utilizando o conteúdo armazenado em variáveis)!

Saída de dados: conteúdo de variáveis

Para imprimir o conteúdo de variáveis, temos a seguinte sintaxe:

são dois argumentos,
separados por vírgula...

```
1 printf( "Mensagem %d", var );
```



Saída de dados: conteúdo de variáveis

Para imprimir o conteúdo de variáveis, temos a seguinte sintaxe:

```
1 printf( "Mensagem %d", var );
```

nome da variável cujo
conteúdo será impresso.

Não tem o & aqui!
O motivo? Veremos no futuro!

Saída de dados: conteúdo de variáveis

Para imprimir o conteúdo de variáveis, temos a seguinte sintaxe:

```
1 printf( "Mensagem %d", var );
```

Na mensagem, aparece também o especificador de formato, o qual deve ser compatível com o tipo da variável que terá seu conteúdo impresso:

%d para int


%f para float

Saída de dados

É possível imprimir o conteúdo de quantas variáveis forem necessárias! Para isso, basta inserir os especificadores de formato adequados nos locais desejados (ou seja, onde o conteúdo da variável deve aparecer). Após a vírgula, colocar os nomes das variáveis na ordem em que devem ser impressas. Exemplo:

nome das variáveis cujos
conteúdos serão impressos
(separados por vírgula).


```
1 printf( "A idade eh %d e a altura %f", idade, altura );
```



Saída de dados

É possível imprimir o conteúdo de quantas variáveis forem necessárias! Para isso, basta inserir os especificadores de formato adequados nos locais desejados (ou seja, onde o conteúdo da variável deve aparecer). Após a vírgula, colocar os nomes das variáveis na ordem em que devem ser impressas. Exemplo:

```
1 printf( "A idade eh %d e a altura %f", idade, altura );
```



especificadores de formato, os quais devem ser compatíveis com o tipo (e a ordem) das variáveis que terão seu conteúdo impresso.

No caso do exemplo, como a primeira variável é um inteiro (idade), o primeiro especificador é %d

Formatação da saída

Código	Função
%d	Escreve um inteiro na tela
%<numero>d	Escreve um inteiro na tela, preenchendo com espaços à esquerda para que ele ocupe pelo menos <numero> casas na tela
%0<numero>d	Escreve um inteiro na tela, preenchendo com zeros à esquerda para que ele ocupe pelo menos <numero> casas na tela
%f	Escreve um float na tela
%.<decimais>f	Escreve um float na tela, arredondando para <decimais> casas decimais
%e	Escreve um float na tela, em notação científica

Exercícios de fixação

Exercício de fixação 1

Faça um programa que imprima na tela o conteúdo de três variáveis do tipo float e duas do tipo int, cujos valores foram lidos do teclado usando scanf().

```
1  #include<stdio.h>
2  int main(){
3
4      float peso, alt, taxa;
5      int idade,codigo;
6
7
8      scanf("%d", &codigo);
9      scanf("%d %f %f", &idade, &peso, &alt);
10     scanf("%f", &taxa);
11
12     printf("Codigo: %d \n");
13     printf("Idade:%d \nPeso:%f \nAltura:%f\n", idade, peso, alt);
14     printf("Taxa: %f", taxa);
15     return 0;
16 }
```

Exercício de fixação 1

Faça um programa que imprima na tela o conteúdo de três variáveis do tipo float e duas do tipo int, cujos valores foram lidos do teclado usando scanf().

```
1  #include<stdio.h>
2  int main(){
3
4      float peso, alt, taxa;
5      int idade,codigo;
6
7
8      scanf("%d", &codigo);
9      scanf("%d %f %f", &idade, &peso, &alt);
10     scanf("%f", &taxa);
11
12     printf("Codigo: %d \n");
13     printf("Idade:%d \nPeso:%f \nAltura:%f\n", idade, peso, alt);
14     printf("Taxa: %f", taxa);
15     return 0;
16 }
```

Exercício de fixação 2

Faça um programa que leia do teclado três inteiros, representando horas, minutos e segundos. Depois, imprima formatado na tela. Por exemplo, ao ler 7, 20 e 9, imprima Horário: 07:20:09. Além disso, imprima um valor float com apenas duas casas decimais.

```
1  #include<stdio.h>
2  int main(){
3
4      int hora, min, seg;
5      float aux = 5.5;
6
7      printf("Digite hora, minuto e segundo: ");
8      scanf("%d %d %d", &hora, &min, &seg);
9
10     printf("Horario: %02d:%02d:%02d\n", hora, min, seg);
11     printf("Float com duas casas: %.2f", aux);
12     return 0;
13 }
```

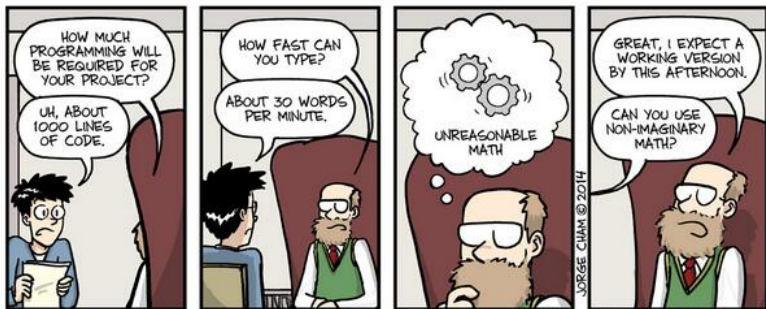
Exercício de fixação 2

Faça um programa que leia do teclado três inteiros, representando horas, minutos e segundos. Depois, imprima formatado na tela. Por exemplo, ao ler 7, 20 e 9, imprima Horário: 07:20:09. Além disso, imprima um valor float com apenas duas casas decimais.

```
1  #include<stdio.h>
2  int main(){
3
4      int hora, min, seg;
5      float aux = 5.5;
6
7      printf("Digite hora, minuto e segundo: ");
8      scanf("%d %d %d", &hora, &min, &seg);
9
10     printf("Horario: %02d:%02d:%02d\n", hora, min, seg);
11     printf("Float com duas casas: %.2f", aux);
12     return 0;
13 }
```

Tarefa

Neste tópico, estudamos entrada e saída de dados! Além dos exercícios de fixação propostos, faça também a lista de exercícios disponibilizada. Ela é simples e rápida, mas vai ajudar você a fixar os conceitos apresentados!



WWW.PHDCOMICS.COM