



Engenharia de Computação

Fundamentos de Programação

Aula 05 – Entrada de Dados

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

Entrada de Dados

- Função **scanf** da biblioteca **stdio.h**
 - Permite ao usuário fornecer dados pelo teclado para o programa.
- **Sintaxe:**

```
scanf("%expressão de controle",&destino);
```

- **Expressão de controle:** "%formato" , parecido com printf...
 - **Exemplo:** "%i" para inteiro, "%f" para float, "%c" para carácter
- **Destino:** &nome_variável
 - **Exemplo:**

```
scanf("%i",&variavel_inteira);  
scanf("%f",&variavel_float);  
scanf("%lf",&variavel_double);  
scanf("%c",&variavel_char);
```

Entrada de Dados

- Exemplo:

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int idade;
    printf("Digite sua idade:");
    scanf("%i",&idade);
    printf("Você possui %i anos\n",idade);
    return 0;
} //main
```

O símbolo &

- Toda variável é associada a um **endereço de memória**
- Ao utilizar uma variável, acessa-se o seu conteúdo localizado no endereço de memória
- O **&** serve para obter o endereço de memória associado a uma variável

```
int idade;  
  
printf("Endereço de memória: %p\n",&idade);  
printf("Valor armazenado: %i\n",idade);  
printf("Digite a idade:");  
scanf("%i",&idade);  
printf("Endereço de memória: %p\n",&idade);  
printf("Novo valor armazenado: %i\n",idade);
```



A função **scanf** recebe o endereço memória

Lendo diversos valores

- Para ler diversos valores em um único scanf
 - coloque os especificadores de tipo na expressão de controle
 - liste os destinos da ordem correspondente
 - Exemplo:
 - O programa abaixo lê: um número inteiro, seguido de uma barra, um número inteiro, seguido de uma barra, um número inteiro.

```
int dia,mes,ano;  
  
printf("Digite a data (dia/mês/ano):");  
scanf("%d/%d/%d",&dia,&mes,&ano);  
  
printf("Dia: %d Mês %d Ano: %d\n",dia,mes,ano);
```

Problemas com o buffer

- Observação sobre o **scanf** para ler caractere:

```
char letra;  
int numero1;  
int numero2;  
  
printf("Digite o numero1:");  
scanf("%i",&numero1);  
printf("Digite uma letra:");  
scanf("%c",&letra);  
printf("Digite o numero2:");  
scanf("%i",&numero2);
```

- Nesse caso, a variável letra não será apresentada para o usuário digitar.
- Por que?
 - O sistema recebe todos os dados em uma memória temporária chamada buffer;
 - A tecla **Enter** do **scanf** na variável numero1, fica no buffer e será armazenada na variável “letra”, pois considera como um caractere qualquer...
- Como resolver isso ??

Problemas com o buffer

- Utilize `setbuf`

```
char letra;  
int numero1;  
int numero2;  
  
printf("Digite o numero1:");  
scanf("%i",&numero1);  
setbuf(stdin,NULL); ←  
printf("Digite uma letra:");  
scanf("%c",&letra);  
printf("Digite o numero2:");  
scanf("%i",&numero2);
```

- Como funciona?

- `setbuf` apaga o buffer do teclado (escrevendo NULL), eliminando qualquer valor que estiver armazenado.

- Pergunta:

- Se colocarmos para ler a variável “letra” primeiro, antes de qualquer `scanf`, então não haveria problema. Por que ?

Exercícios

1) Faça um programa que receba a altura de uma pessoa e calcule e exiba seu peso ideal utilizando a seguinte:

$$\text{peso ideal} = 72.7 \times \text{altura} - 58$$

2) Faça um programa que calcule e mostre a idade de uma pessoa a partir do ano em que a pessoa nasceu e do ano atual. Os valores serão digitados pelo usuário.

Obs.: Considere que ela já fez aniversário neste ano.

3) Tendo como entrada de dados o total vendido por um funcionário no mês, calcule a sua comissão e o salário bruto no mês. Para isso, considere um salário base de R\$1.200,00 e comissão de 10% sobre o total vendido.

Exercícios

- 4) Calcule a média aritmética de 4 números reais que o usuário digitar. Imprima a média na tela apenas com 2 casas decimais.
- 5) Sabendo que a função `sqrt(valor)`, que está na biblioteca `math.h` retorna a raiz quadrada do *valor*, calcule a raiz quadrada de um número que o usuário digitar.



Utilizar tipo **double** para números reais nas funções matemáticas!
No printf ou scanf, utilizar **%lf**