

CURSO SUPERIOR DE ADS

Estruturas de Decisão



Prof. Fernando Marlon Soares Figueiredo
Disciplina: Programação Estruturada



Aula de Hoje

- Introdução às Estruturas de Decisão
- Operadores lógicos: &&, || e !
- Estrutura de seleção simples: if / else
- Estrutura de seleção múltipla: switch / case
- Aplicar muitos exercícios

Introdução às Estruturas de Decisão

- Até então, todos os programas construídos até aqui seguiam apenas um fluxo de execução, podendo apenas ter seu resultado modificado com uma entrada diferente do usuário.
- A partir de agora vamos aprender a construir programas que executam trechos de códigos apenas se determinadas condições forem atingidas.

Operadores Relacionais e Lógicos

- Além dos operadores aritméticos, existem outros dois tipos de operadores fundamentais para se trabalhar com condições. Enquanto os **operadores relacionais** são responsáveis por comparar expressões, os **operadores lógicos** permitem combinar operadores relacionais para construção de condições mais complexas.
- Ambos os operadores ao serem computados resultam em um único valor do qual pode ser verdadeiro ou falso.
- Valores verdadeiros costumam ser associados ao número 1 (ou alternativamente, outro número qualquer diferente de zero) e falsos estão associados com zero.

Operadores Relacionais

São operadores que relacionam duas variáveis, constantes ou literais, resultando em um valor booleano (verdadeiro ou falso).

Operador	Significado
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
==	Igual
!=	Diferente

Operadores Relacionais

1. $3 < 5$: Verdadeiro

2. $0 \neq 4$: Verdadeiro

3. $15 == 2$: Falso

4. $3.14 == \text{PI}$: Falso

5. $x \leq 3$: ?

- 1 a 3 são operações triviais.
- 4 é falso porque PI é uma constante infinita, de tamanho maior que 3,14.
- 5 vai depender do valor que x assume no programa.

Operadores Lógicos

- Agem diretamente sobre expressões cujo resultado é booleano (ou lógico), transformando seus valores. Os valores são alterados de acordo com a função lógica de cada operador.
- Atente que os operadores E e OU são binários (aplicáveis sobre duas expressões), enquanto o NÃO é unário (aplicável sobre uma única expressão).

Operador	Significado
&&	E lógico
	OU lógico
!	NÃO lógico

Operador E

Valor 1	Valor 2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

Operador OU

Valor 1	Valor 2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

Operador NÃO

Valor 1	Resultado
V	F
F	V

Precedência de operadores relacionais e lógicos

- Operadores relacionais (`==`, `>`, `<`, `>=`, `<=`, `!=`) possuem maior precedência do que os operadores lógicos (`&&`, `||` e `!`).
- As expressões a seguir são equivalentes:

1. `x == 4 || y > -1`

2. `((x == 4) || (y > -1))`

Exemplos de Expressões Lógicas e Relacionais

1. $(2 < 3) \ \&\& (1 < 3)$: (V) E (V): Verdadeiro
2. $(2 \leq 3) \ || \ (4 < 3)$: (V) OU (F): Verdadeiro
3. $!(3 \neq 4)$: NÃO (V): Falso
4. $(1 \ || \ (4 == 3)) \ \&\& 0$: (V) E (F): Falso
5. $x \leq 3 \ \&\& y == 4$: ?

Seleção Simples

if/else

Sintaxe do if /else

- A estrutura permite que o programa altere seu fluxo de execução caso determinadas condições sejam alcançadas.
- A condição resulta em uma expressão cujo resultado é um valor lógico.

```
if(condicao)
{
    // Bloco de codigo (condicao eh verdadeira)
}

else
{
    // Bloco de codigo (condicao eh falsa) (opcional)
}
```

Exercício a

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 11;  
if (i > 10)  
    i++;  
    i++;  
printf(i);
```

Exercício a

Resposta: 13

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 11;  
if (i > 10)  
    i++;  
    i++;  
printf(i);
```

Exercício b

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 10;  
if (i > 10)  
    i++;  
    i++;  
printf(i);
```

Exercício b

Resposta: 11

Qual o valor irá aparecer no console de acordo com o trecho de código abaixo?

```
int i = 10;  
if (i > 10)  
    i++;  
i++;  
printf(i);
```

Fique Atento!

- Quando não delimitamos o escopo dos comandos que irão ser executados com as chaves, só fará parte do escopo do bloco Verdadeiro (ou Falso) do if apenas a linha de código seguinte. Apenas ela.
- Dessa forma, as duas linhas de código a seguir terão o mesmo resultado:

```
if (i > 10) i++;
```

```
if (i > 10) { i++; }
```

Exercício 1

Faça um programa que receba a idade do usuário e informa se ele é maior ou menor de idade.

Digite sua idade: 18

Maior de idade!

Exercício 1 - Resposta

Note que, em geral, quando blocos condicionais possuem apenas um único comando dentro deles, o uso de chaves é opcional. Portanto, tanto as linhas 11 e 13 quanto as linhas 15 e 17 do Programa 3.2 (que contém abre e fecha chaves) poderiam deixar de existir nesse caso.

```
1 #include <stdio.h>
2
3 int main ()
4 {
5     int idade;
6
7     printf("Digite sua idade:");
8     scanf("%d", &idade);
9
10    if(idade < 18)
11    {
12        printf("Menor de idade!\n");
13    }
14    else
15    {
16        printf("Maior de idade!\n");
17    }
18
19    return 0;
20 }
```

Exercício 2

Modifique o exercício anterior para classificar o usuário em criança (menor que 12 anos), adolescente (entre 12 e 18 anos) e adulto (maior que 18 anos).

Digite sua idade: 25

Adulto!

Exercício 2 - Resposta

```
3 int main ()
4 {
5     int idade;
6
7     printf("Digite sua idade:");
8     scanf("%d", &idade);
9
10    if(idade < 12)
11    {
12        printf("Crianca!\n");
13    }
14    else if(idade >= 12 && idade <= 18)
15    {
16        printf("Adolescente!\n");
17    }
18    else
19    {
20        printf("Adulto!\n");
21    }
22
23    return 0;
24 }
```

Exercício 3

Faça um programa que peça ao usuário para digitar um número inteiro e indique se o número digitado é positivo, negativo ou nulo.

Exercício 3 - Resposta

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero;
5      printf("Digite um número\n");
6      scanf("%d", &numero);
7
8      if (numero>0) {
9          printf("positivo");
10     } else if (numero<0) {
11         printf("negativo");
12     } else {
13         printf("nulo");
14     }
15     return 0;
16 }
```

Exercício 4

Peça ao usuário para inserir um número inteiro e verifique se ele é par ou ímpar. Imprima uma mensagem indicando o resultado.

Exemplo de Saída

Digite um número inteiro: 7

O número 7 é ímpar.

Exercício 4 - Resposta

```
1  #include <stdio.h>
2
3  v int main(void) {
4      int numero;
5      printf("Digite um número\n");
6      scanf("%d", &numero);
7
8  v  if (numero%2==0) {
9      printf("Número par");
10 v } else {
11     printf("Número ímpar");
12 }
13 return 0;
14 }
```

Exercício 5

Faça um programa que peça ao usuário para digitar três números reais e imprima o maior número digitado. Caso os valores sejam iguais, qualquer um deles pode ser impresso.

Exercício 5 - Resposta

```
1  #include <stdio.h>
2
3  int main(void) {
4      float n1, n2, n3;
5      printf("Digite um número\n");
6      scanf("%f", &n1);
7      printf("Digite outro número\n");
8      scanf("%f", &n2);
9      printf("Digite mais um número\n");
10     scanf("%f", &n3);
11
12     if (n1>n2 && n1>n3) {
13         printf("O maior é: %.2f", n1);
14     } else if (n2>n1 && n2>n3) {
15         printf("O maior é: %.2f", n2);
16     } else if (n3>n1 && n3>n2) {
17         printf("O maior é: %.2f", n3);
18     } else {
19         printf("Os números são iguais %.2f", n1);
20     }
21     return 0;
22 }
```

Exercício 6

- Faça um programa que pergunte ao usuário qual figura bidimensional ele deseja saber a área: círculo ou retângulo. Esta opção deve ser registrada em um inteiro, com um valor para círculo e outro para retângulo, sendo informado ao usuário.
- Caso o círculo seja escolhido, peça para o usuário informar o raio.
- Caso o retângulo seja escolhido, peça para o usuário digitar a base e a altura. Em seguida informe o valor da área, dependendo da figura escolhida.
- Utilize PI como uma constante no valor de 3.1315.
- Área do círculo é $PI * raio^2$

Exercício 6 - Resposta

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void) {
5      int opcao, base, altura;
6      float raio, area;
7      printf("Cálculo de Área\n");
8      printf("Digite 1 para círculo, 2 para retângulo\n");
9      scanf("%d", &opcao);
10
11     if (opcao == 1) {
12         printf("Área do Círculo\n");
13         printf("Qual o raio?\n");
14         scanf("%f", &raio);
15         //Função pow(base, expoente)
16         area = 2 * M_PI * pow(raio, 2);
17         printf("Área: %.2f", area);
```

```
18
19     } else if (opcao == 2) {
20         printf("Área do Retângulo\n");
21         printf("Qual a base?\n");
22         scanf("%d", &base);
23         printf("Qual a altura?\n");
24         scanf("%d", &altura);
25         area = base * altura;
26         printf("Área: %.2f", area);
27     } else {
28         printf("Alternativa inválida");
29     }
30     return 0;
31 }
```

Seleção Múltipla

switch case

Seleção Múltipla

- Diferente da seleção simples (if/else), a seleção múltipla exerce o controle sobre uma única variável, especificando diretamente o que ocorre para cada valor que essa variável pode assumir.
- Esta variável pode ser do tipo int ou char e seu uso é bem mais restrito do que seleção simples.
- Comandos de seleção múltipla devem ser utilizados quando possível para que o código fique mais bem organizado.

Sintaxe do comando switch/case

```
switch(variavel)
{
    case 0:
        // código executado caso a variável seja 0
        break;
    case 1:
        // código executado caso a variável seja 1
        break;
    ...
    case N:
        // código executado caso a variável seja N
        break;
    default:
        // código executado se nenhum case for satisfeito
}
```

Sintaxe do comando switch/case

- Os cases não podem ser variáveis, apenas literais e/ou constantes referentes a valores possíveis da variável do switch.
- Podem haver quantos cases forem necessários, desde que respeitem as regras mencionadas.
- O comando break deve obrigatoriamente ser usado ao final de cada cláusula, a fim de que as demais não sejam executadas.

Exemplo

```
1  #include <stdio.h>
2  int main ()
3  {
4      char dir;
5      printf("Digite uma direcao (w, a, s, d):");
6      scanf("%c", &dir);
7      switch(dir)
8      {
9          case 'w':
10             printf("Cima");
11             break;
12             case 'a':
13                 printf("Esquerda");
14                 break;
15                 case 's':
16                     printf("Baixo");
17                     break;
18                     case 'd':
19                         printf("Direita");
20                         break;
21                     default:
22                         printf("Comando invalido");
23                 }
24             return 0;
25 }
```

Exercício 7

Faça um programa que peça ao usuário para digitar um número inteiro de 1 a 4 (inclusive) e receba uma mensagem personalizada de acordo com o número digitado.

Exercício 7 - Resposta

```
4  int main(void) {  
5      int opcao;  
6      printf("Escolha de 1 a 4 para uma mensagem:");  
7      scanf("%d", &opcao);  
8      switch(opcao) {  
9          case 1:  
10         printf("Bom dia");  
11         break;  
12         case 2:  
13         printf("Boa tarde");  
14         break;  
15         case 3:  
16         printf("Boa noite");  
17         break;  
18         case 4:  
19         printf("Vá dormir");  
20         break;  
21         default:  
22         printf("Opção inválida");  
23     }  
24     return 0;  
25 }
```

Exercício 8

- Faça um programa que receba o valor de uma compra e um inteiro indicando o tipo do produto internacional. Se o tipo digitado for 0 indica que deve ser cobrada taxa de 10% de seu valor.
- Caso o tipo digitado seja qualquer outro número, indica que deve ser cobrada taxa de 20% de seu valor. Ao final, indique qual o valor total do produto a ser pago pelo usuário.

Exercício 9

- Elabore um programa que leia dois valores do usuário e a operação que ele deseja executar (Operações: soma, subtração, divisão, multiplicação).
- Execute a operação desejada e imprima na tela.

Exercício 10

Dado o valor do produto e a forma de pagamento.

1. à vista;
2. à prazo.

Se o produto for pago à vista aplique um desconto de 10% antes de mostrar o valor final, senão informe o mesmo valor do produto.

Exercício 11

- Uma loja fornece 10% de desconto para funcionários e 5% de desconto para clientes vips.
- Faça um programa que calcule o valor total a ser pago por uma pessoa. O programa deverá ler o valor total da compra efetuada e um código que identifique se o comprador é um cliente comum (1), funcionário (2) ou vip (3).

Exercício 11 - Resposta

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void) {
5      int opcao;
6      float total;
7      printf("Qual o valor total da compra?\n");
8      scanf("%f", &total);
9      printf("Você é: (1) cliente normal, (2) funcionário, (3) cliente vip?\n");
10     scanf("%d", &opcao);
11     switch(opcao) {
12         case 1:
13             printf("Cliente normal");
14             printf("0 total é: R$ %.2f", total);
15             break;
```

```
16         case 2:
17             printf("Funcionário");
18             total*=0.9;
19             printf("0 total é: R$ %.2f", total);
20             break;
21         case 3:
22             printf("Cliente Vip");
23             total*=0.95;
24             printf("0 total é: R$ %.2f", total);
25             break;
26         default:
27             printf("Opção inválida");
28     }
29     return 0;
30 }
```

Exercício 12

- A taxa de natalidade e a taxa de mortalidade são indicadores estatísticos de fundamental importância por ajudar a compreender aquilo que os especialistas chamam de dinâmica populacional e a entender sua relação com variáveis que influenciam o desenvolvimento, como qualidade de vida, migrações, fatores socioeconômicos e localização. Sabendo disso, crie um programa que calcule os dois indicadores utilizando o comando switch.

A taxa de natalidade é calculada pela seguinte fórmula:

- taxa de natalidade = $(n^{\circ} \text{ de crianças nascidas} \times 1000) / n^{\circ} \text{ de habitantes}$
- taxa de mortalidade = $(n^{\circ} \text{ de pessoas mortas} \times 1000) / n^{\circ} \text{ de habitantes}$

Exercício 12 - Resposta

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void) {
5      int opcao, habitantes, nascidos, morridos;
6      float total;
7      printf("Qual o total de habitantes?\n");
8      scanf("%d", &habitantes);
9      printf("Você quer saber: (1) natalidade ou (2) mortalidade?\n");
10     scanf("%d", &opcao);
11     switch(opcao) {
12         case 1:
13             printf("Natalidade\n");
14             printf("Quantos nasceram?\n");
15             scanf("%d", &nascidos);
16             total = ((nascidos * 1000) / habitantes );
17             printf("A natalidade é %.2f", total);
18             break;
```

```
19         case 2:
20             printf("Mortalidade\n");
21             printf("Quantos morreram?\n");
22             scanf("%d", &morridos);
23             total = ((morridos * 1000) / habitantes );
24             printf("A natalidade é %.2f", total);
25             break;
26         default:
27             printf("Opção inválida");
28     }
```

Exercício 13

- Faça um programa de conversão de base numérica. O programa deverá apresentar uma tela de entrada com as seguintes opções:

< Conversão de base >

1: decimal para hexadecimal

2: hexadecimal para decimal

3: decimal para octal

4: octal para decimal

5: Encerra

Exercício 13 - continuação

- Lembre que, você deve informar as opções para que a partir da opção escolhida, o programa deva pedir o número na base escolhida, lê-lo e apresentá-lo na base desejada.
- Em seguida, o programa deve perguntar ao usuário se ele deseja retornar ao menu principal ou finalizar o programa.
- Obs: O problema pode ser estendido a outras bases, de acordo com o interesse do aluno.