

Fundação Hermínio Ometto

Bacharelado em Sistemas de Informação

SIF009 - Linguagem de Programação I

Prof. Dr. Sérgio Luis Antonello

Aula 03

18/08/020

Sumário da aula

Primeiro momento (revisão)

- ✓ Tipos de dados
- ✓ Operadores
- ✓ Formatação dos dados das funções de entrada e saída.

Segundo momento (conteúdo)

- ✓ Estruturas de decisão/seleção
 - ✓ if – else
 - ✓ switch - case

Terceiro momento (síntese)

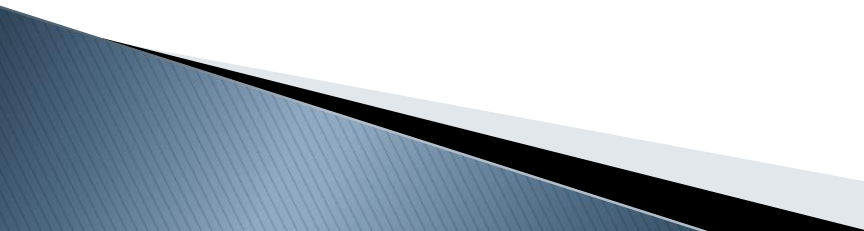
- ✓ Retome pontos importantes da aula

1. Primeiro momento: revisão

Os tipos de dados básicos em linguagem C são:

- ▶ **char**: para armazenar um caractere.
- ▶ **int**: número inteiro.
- ▶ **float**: número em ponto flutuante de precisão simples (números reais).
- ▶ **double**: número em ponto flutuante de precisão dupla.

A linguagem C possui modificadores que podem alterar os tipos básicos acima.



1. Primeiro momento: revisão

Precedência de operadores (quem executa primeiro).

Operador	Tipo de operação
postfix ++ e postfix --	
prefix ++ e prefix --	
* / %	Mult,. Divi. Resto div.
+ -	Adição e Subtração
== <> < > <= >=	Relacional
!	Negação
&&	Lógico E
	Lógico OU

1. Primeiro momento: revisão

Formatações mais usadas em I/O

- %d inteiro decimal
- %i inteiro
- %f ponto flutuante (float)
- %lf ponto flutuante longo (double)
- %c caractere simples
- %s cadeia de caractere

1. Primeiro momento: revisão

▶ Exemplos

- `printf("mensagem");`
- `printf("mensagem\n");` // avança para próxima linha

▶ Formatação para tipo int

```
int qde = 18;
```

```
printf("Quantidade: %d peças.", qde);
```

- ## ▶ Saída correspondente
- Quantidade: 18 peças.

Também pode ser usado %i para inteiro

1. Primeiro momento: revisão

- ▶ Formatação para tipo float
float nota1 = 8.0;
float nota2 = 9.0;
printf("N1: %f; N2: %.1f; Média: %.2f", nota1,
nota2, (nota1+nota2)/2);
- ▶ Saída correspondente
N1: 8.000000; N2: 9.0; Média: 8.50

1. Primeiro momento: revisão

Outras formatações

- ▶ int: %d ou %i
- ▶ long int: %ld ou %li
- ▶ long long int: %lld ou %lli

- ▶ float: %f
- ▶ double: %lf
- ▶ long double: %Lf

- ▶ caractere: %c
- ▶ cadeia de caracteres: %s

- ▶ ponteiro: %p

1. Primeiro momento: revisão

- ▶ Outras formatações
 - ▶ \n nova linha
 - ▶ \t - imprime uma tabulação
 - ▶ %% - imprima o caractere %
 - ▶ \b retrocesso
 - ▶ \" aspas
 - ▶ \f salta formulário
 - ▶ \0 nulo

1. Primeiro momento: revisão

Correção de exercícios



2. Segundo momento

- ▶ Estruturas de decisão (seleção)
 - if
 - switch case



2. Segundo momento: motivação

Um aluno do curso de Especialização em Engenharia de Software da FHO desenvolveu seu TCC com tema “Uso de jogos no ensino e aprendizagem de Engenharia de Software”.

Uma parte do TCC é ler e analisar artigos publicados em repositórios digitais de artigos científicos. Sendo assim, usando motores de busca na internet, a partir de termos como “jogos” e “engenharia de software” ele encontrou pouco mais de 3.100 artigos.

Acontece que, como é de conhecimento de todos, os resultados de busca na internet quase sempre trazem muitos resultados indesejados.

Para facilitar seu trabalho no sentido de ler apenas trabalhos mais relevantes, baseado em um modelo de pesquisa realizado anteriormente, ele pensou em selecionar apenas trabalhos em que alguns termos como “ensino” e “aprendizagem” aparecessem pelo menos 10 vezes em cada texto encontrado na web.

Como poderia ser operacionalizado esse modelo “pensado”?

3. Comando if-else

O comando **if** instrui o computador a tomar uma **decisão**

Sintaxe:

```
if (expressão de teste)  
    comando a ser executado;
```

Se a expressão de teste for verdadeira, o comando if executa a instrução. Caso contrário a instrução não será executada.



3. Comando if-else

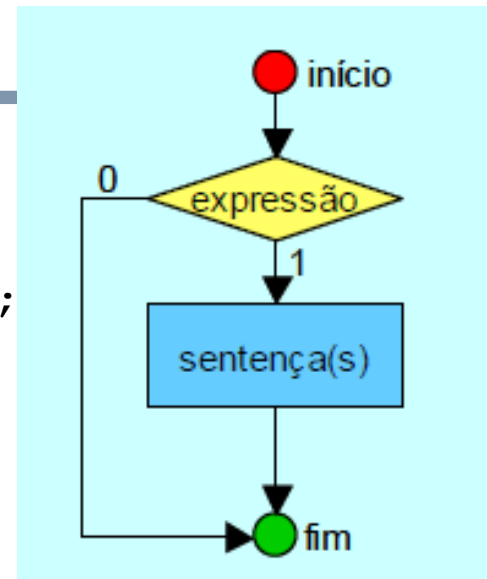
O **else**, quando associado ao **if**, executará uma instrução (ou instruções) se a expressão de teste do comando if for **falsa**.

Sintaxe:

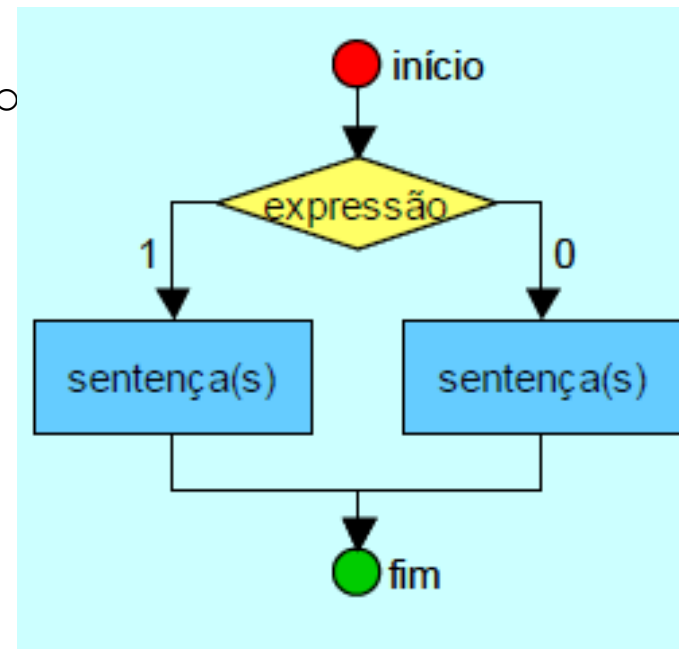
```
if (expressão de teste) {  
    Comandos a serem executados;  
}  
else {  
    Comandos a serem executados;  
}
```

3. Comando if-else

```
if (nota > 10) {  
    printf ("Nota inválida\n");  
    printf ("Nota não pode ser maior que dez");  
}
```



```
if (nota >= 5) {  
    printf ("Parabéns, você passou de ano");  
}  
else {  
    printf ("\n\nVocê foi reprovado");  
}
```



3. Comando if-else

O encadeamento ou aninhamento de comandos ocorre quando um comando está dentro de outro comando de mesmo tipo.

Um comando if pode estar dentro de outro comando if. Dizemos que o if interno está **aninhado**.

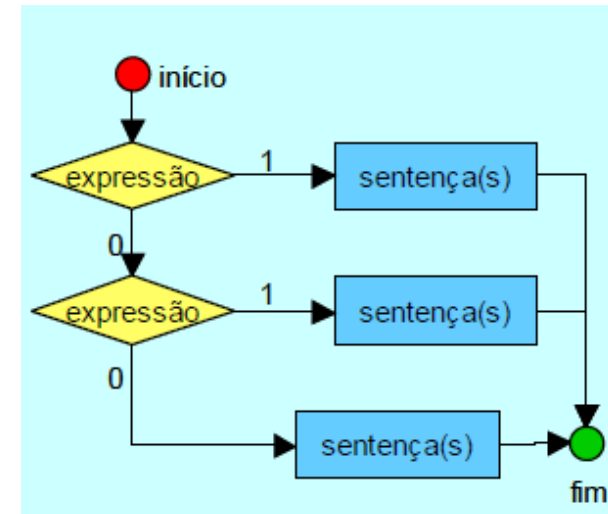
Sintaxe:

```
if (expressão de teste 1){  
    if (expressão de teste 2) {  
        Comandos a serem executados;  
    }  
}
```


3. Comando if-else

if aninhado.

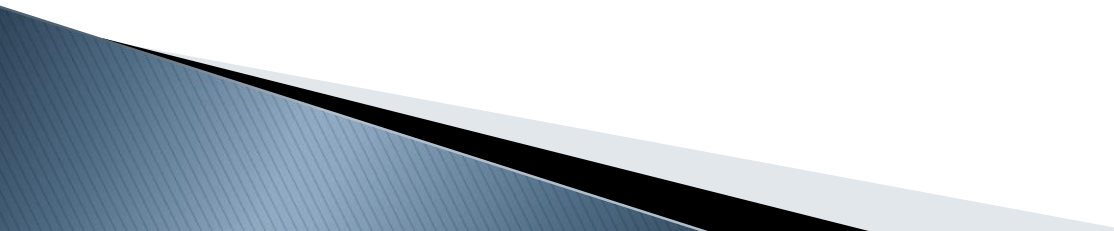
```
if (idade < 18) {  
    printf("Nao pode ter habilitação");  
}  
else {  
    if (idade < 65) {  
        printf("Renovação da CNH a cada 5 anos");  
    }  
    else  
    {  
        printf(" Renovação da CNH a cada 3 anos");  
    }  
}
```



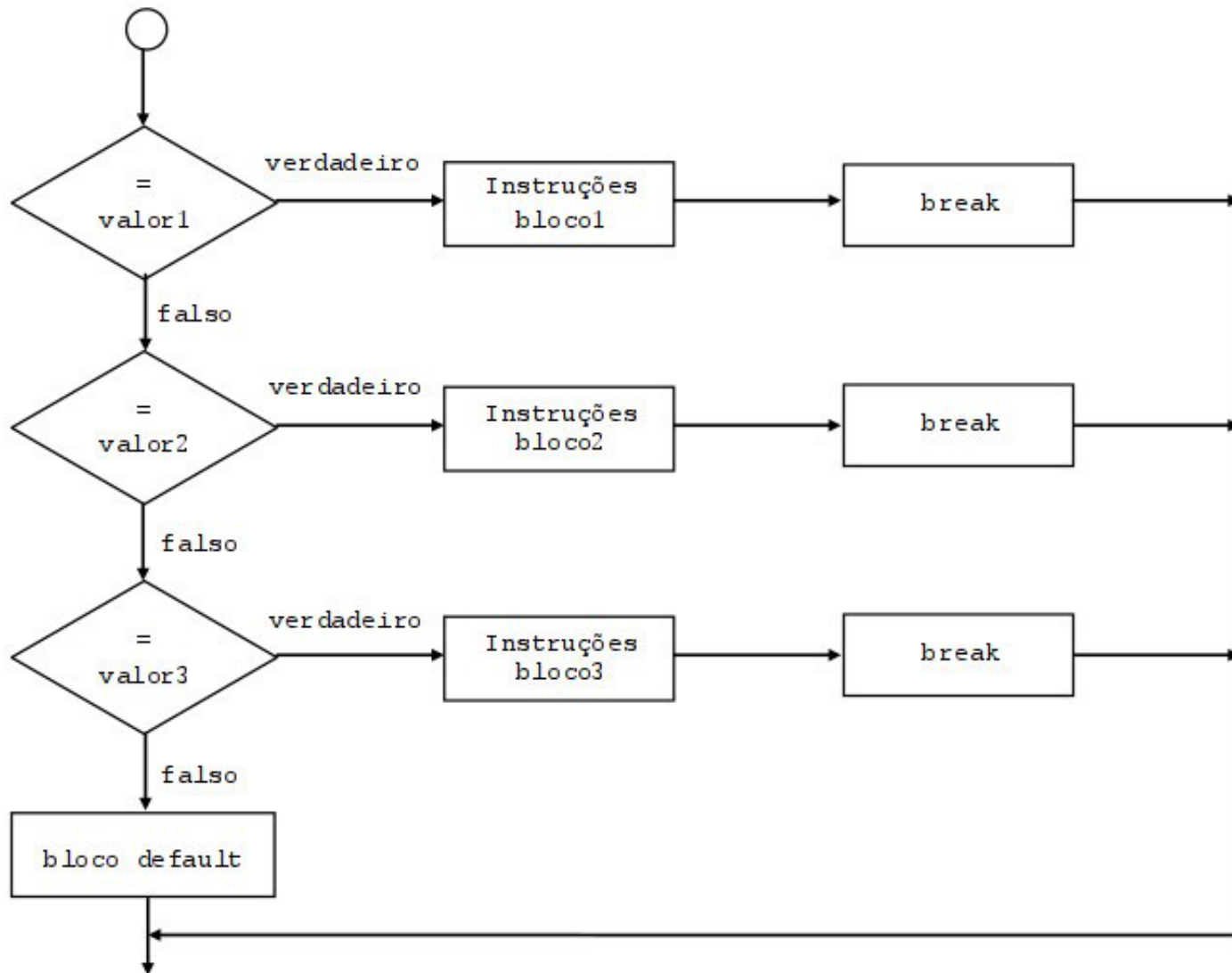
3. Comando if-else

```
1  #include <stdio.h>
2  #include <conio.h>
3  int main (void )
4  {
5      int valor;
6
7      printf ("Digite um valor de 1 a 7: ");
8      scanf ("%d", &valor);
9
10
11     if (valor == 1)
12         printf ("Domingo\n");
13     else
14         if (valor == 2)
15             printf ("Segunda\n");
16         else
17             if (valor == 3)
18                 printf ("Terça\n");
19             else
20                 if (valor == 4)
21                     printf ("Quarta\n");
22                 else
23                     if (valor == 5)
24                         printf ("Quinta\n");
25                     else
26                         if (valor == 6)
27                             printf ("Sexta\n");
28                         else
29                             if (valor == 7)
30                                 printf ("Sabado\n");
31                             else
32                                 printf ("Valor invalido!\n");
33
34     getch();
35     return 0;
36 }
```

4. Comando switch-case

- O comando **switch-case** também é uma estrutura de seleção/decisão.
 - O comando switch testa sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou caracteres.
 - Quando o valor coincide, os comandos associados àquela constante são executados.
- 

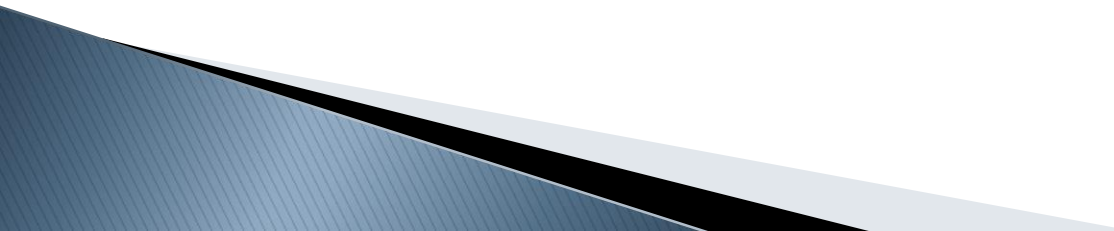
4. Comando switch-case



4. Comando switch-case

O comando **break** causa uma saída imediata do switch. Se não existir um comando break seguindo as instruções de um caso, o programa segue executando todas as instruções dos casos abaixo.

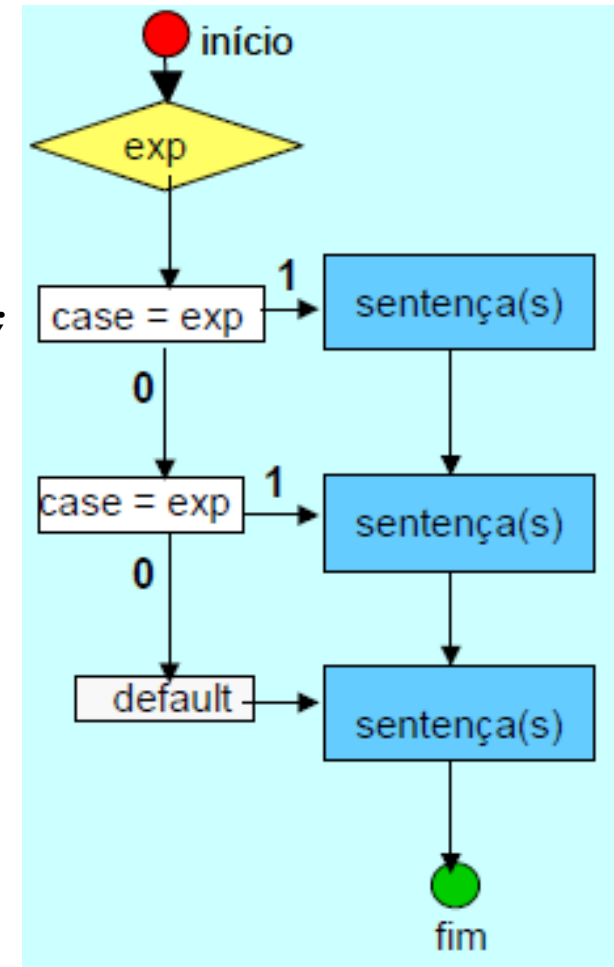
Se nenhum caso for satisfeito e existir um caso **default**, a execução começará nele, senão o programa processará as instruções seguintes ao bloco switch. Um default é opcional.



4. Comando switch-case

Sintaxe:

```
switch (pedido)
{
    case 1:
        printf ("Seu pedido foi X-Salada");
        MontaLanche(1);
        break;
    case 2:
        printf ("Seu pedido foi X-Bacon");
        MontaLanche(2);
        break;
    default:
        printf ("Pedido inválido");
}
```



4. Comando switch-case

```
1  #include <stdio.h>
2  #include <conio.h>
3  int main (void )
4  {
5      int valor;
6
7      printf ("Digite um valor de 1 a 7: ");
8      scanf ("%d", &valor);
9
10     switch ( valor )
11     {
12         case 1 :
13             printf ("Domingo\n");
14             break;
15
16         case 2 :
17             printf ("Segunda\n");
18             break;
19
20         case 3 :
21             printf ("Terça\n");
22             break;
23
24         case 4 :
25             printf ("Quarta\n");
26             break;
27
28         case 5 :
29             printf ("Quinta\n");
30             break;
31
32         case 6 :
33             printf ("Sexta\n");
34             break;
35
36         case 7 :
37             printf ("Sabado\n");
38             break;
39
40         default :
41             printf ("Valor invalido!\n");
42     }
43
44     getch();
45     return 0;
46 }
```

5. Exercícios

**Vamos
Programar!**



5. Exercícios

1) Calculadora simples

- ✓ Faça um programa, em linguagem C, que leia dois números reais num1 e num2, e um número inteiro “op”.
- ✓ O valor da variável “op” determina a operação matemática que deve ser feita com os números inteiros (1- soma, 2- subtração, 3- multiplicação e 4-divisão).
- ✓ O programa deve exibir a operação realizada e o resultado (exemplo: $1 + 1 = 2$).
- ✓ Caso seja digitado um valor para “op” diferente (menor que 1 ou maior que 4) o programa deve exibir a mensagem “Operação inválida”.

OBS: Utilize a estrutura if-else.

5. Exercícios

2) Mais uma calculadora simples

- ✓ Faça outro programa, em linguagem C, considerando o enunciado do exercício anterior, só que agora, utilize a estrutura switch-case.

3) URI 1044 - Múltiplos

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1044>

4) URI 1035 - Teste de Seleção 1

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1035>

5) URI 1052 - Mês

<https://www.urionlinejudge.com.br/judge/pt/problems/view/1052>

6. Terceiro momento: síntese

- As estruturas de decisão (seleção) possibilitam que o programa, durante a sua execução, tome um caminho ou outro, dependendo de uma condição previamente estabelecida.
- Os comandos usados
 - if - else
 - switch - case