

# Introdução à Computação Condicionais Compostas

Prof. Dr. Marcos Paulino Roriz Junior (marcosroriz@ufg.br)



UFG

UNIVERSIDADE  
FEDERAL DE GOIÁS



ENGENHARIA DE  
TRANSPORTES

FCT  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



UFG  
UNIVERSIDADE  
FEDERAL DE GOIÁS

# Condição

- Permite tratar caso falho na condição simples

1. SE **media**  $\geq$  6

2. ENTÃO INÍCIO

3. passo 1;

4. passo 2;

5. passo 3;

6. FIM

= bloco de instruções

7. SE **media**  $<$  6

8. ENTÃO INÍCIO

9. passo 1;

10. passo 2;

11. passo 3;

12. FIM

= bloco de instruções



# Condicionais Compostas

```
if (condição)
    comando1;
else
    comando2;
```

Se a condição for verdadeira, será executado o comando1; se for falsa, será executado o comando2.

```
if (condição)
{
    comando1;
    comando2;
}
else
{
    comando3;
    comando4;
}
```

Se a condição for verdadeira, o comando1 e o comando2 serão executados; caso contrário, o comando3 e o comando4 serão executados.



# Fluxo de execução

- Idéia por traz é simples:
- Se condição for verdadeira executa o bloco 1, senão executa o bloco 2

1. **SE (condição)**

2. **ENTÃO INÍCIO**

3. passo 1;

4. passo 2;

5. passo 3;

6. **FIM**

= bloco 1

7. **SENÃO INÍCIO**

8. passo 1;

9. passo 2;

10. passo 3;

11. **FIM**

= bloco 2



# Condição

1. SE **media**  $\geq$  6

2. ENTÃO INÍCIO

3. escreva "Aprovado" bloco1

4. FIM

5. SENÃO INÍCIO

6. escreva "Reprovado" bloco2

7. FIM

1. ...

2. if (media  $\geq$  6) {

3. printf("aprovado"); bloco1

4. } else {

5. printf("reprovado"); bloco2

6. }

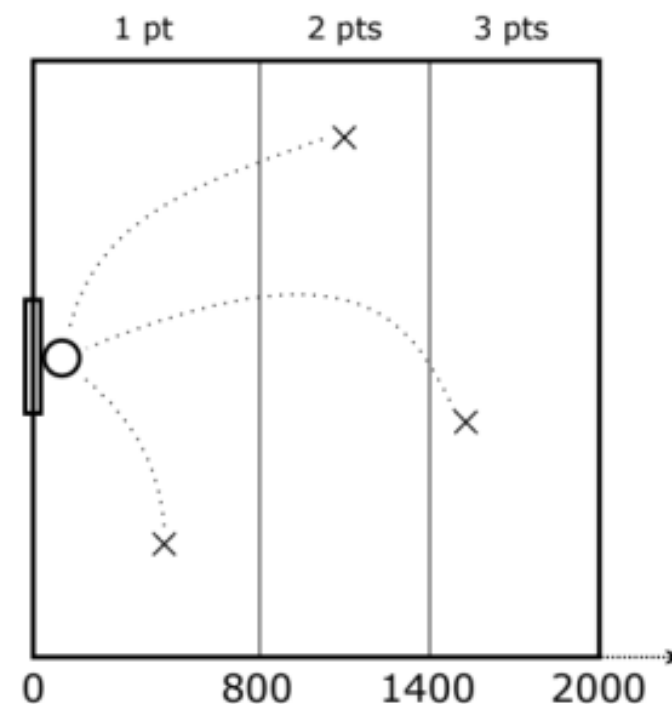


# Condicionais Compostas: Exemplo

Questão 23. (OBI) A organização da OIBR, Olimpíada Internacional de Basquete de Robô, está começando a ter problemas com dois times: os Bit Warriors e os Byte Bulls. É que os robôs desses times acertam quase todos os lançamentos, de qualquer posição na quadra! Pensando bem, o jogo de basquete ficaria mesmo sem graça se jogadores conseguissem acertar qualquer lançamento, não é mesmo?

Uma das medidas que a OIBR está implantando é uma nova pontuação para os lançamentos, de acordo com a distância do robô para o início da quadra. A quadra tem 2000 centímetros de comprimento, como na figura. Dada a distância  $D$  do robô até o início da quadra, onde está a cesta, a regra é a seguinte:

- Se  $D \leq 800$ , a cesta vale 1 ponto;
- Se  $800 < D \leq 1400$ , a cesta vale 2 pontos;
- Se  $1400 < D \leq 2000$ , a cesta vale 3 pontos.



A organização da OIBR precisa de ajuda para automatizar o placar do jogo. Dado o valor da distância  $D$ , você deve:

- (a) escrever um algoritmo em pseudocódigo e um programa para calcular o número de pontos do lançamento.
- (b) desenhe um fluxograma (por ex, utilize o <https://www.lucidchart.com/>) para calcular o número de pontos do lançamento

**Entrada** A primeira e única linha da entrada contém um inteiro  $D$  indicando a distância do robô para o início da quadra, em centímetros, no momento do lançamento.

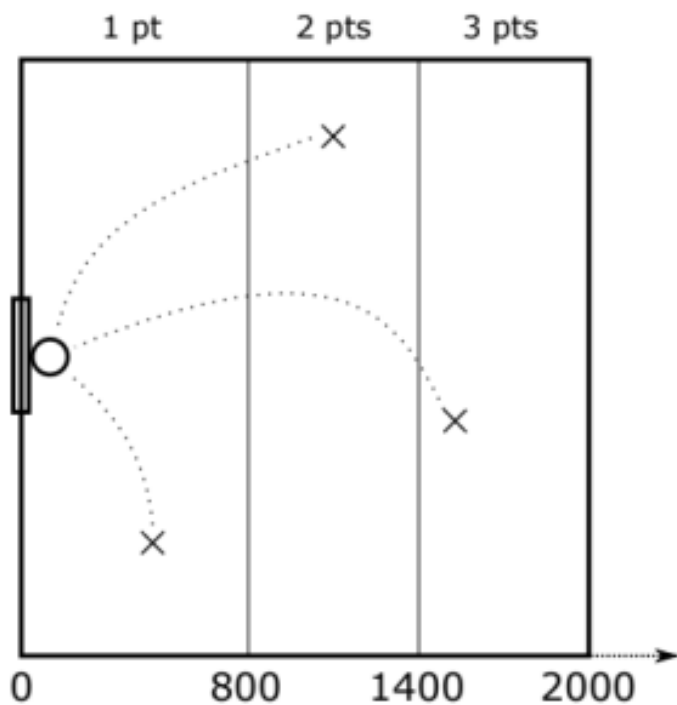
**Saída** Seu algoritmo/fluxograma deve produzir uma única linha, contendo um inteiro, 1, 2 ou 3, indicando a pontuação do lançamento.

**Exemplos de Entrada e Saída.**

Entrada:	Saída:
1720	3
250	1
1400	2



# Condicionais Compostas



```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.     ESCREVA "1 ponto"
7. FIM
8. SENÃO INÍCIO
9.     SE d <= 800 e d <= 1400
10.    ENTÃO INÍCIO
11.        ESCREVA "2 pontos"
12.    FIM
13.    SENÃO INÍCIO
14.        ESCREVA "3 pontos"
15.    FIM
16. FIM
17.FIM_ALGORITMO
  
```

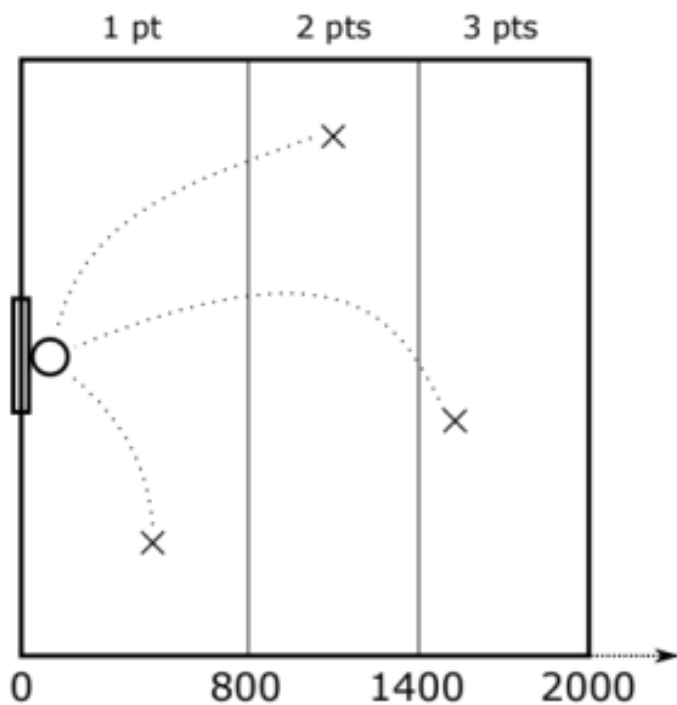
BLOCO 1

BLOCO 2





# Condicionais Compostas



```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.     ESCREVA "1 ponto"
7. FIM
8. SENÃO INÍCIO
9.     SE d <= 800 e d <= 1400
10.    ENTÃO INÍCIO
11.        ESCREVA "2 pontos"
12.    FIM
13.    SENÃO INÍCIO
14.        ESCREVA "3 pontos"
15.    FIM
16. FIM
17. FIM_ALGORITMO
  
```

BLOCO 1

BLOCO 2



# Condicionais encadeadas

- Escreva um algoritmo em pseudocódigo que lê a média  $m$  de um aluno e imprime a sua situação:
- “aprovado *summa cum laude*”, se  $m \geq 9$
- “aprovado *cum laude*”, se  $m \geq 8$
- “aprovado”, se  $m \geq 6$
- “reprovado”, se  $m < 6$



# Condição: Exemplo

1. INÍCIO\_ALGORITMO

2. DECLARE m

3. LEIA m

4. SE m  $\geq$  9

5. ENTÃO INÍCIO

6.     ESCREVA "aprovado summa cum laude"

BLOCO 1

7.     FIM

8. SENÃO INÍCIO

9.     SE m  $\geq$  8

10.     ENTÃO INÍCIO

11.         ESCREVA "aprovado cum laude"

12.         FIM

13.     SENÃO INÍCIO

14.         SE m  $\geq$  6

15.         ENTÃO INÍCIO

BLOCO 2

16.             ESCREVA "aprovado"

17.             FIM

18.         SENÃO INÍCIO

19.             ESCREVA "reprovado"

20.             FIM

21.         FIM

22.     FIM

23. FIM\_ALGORITMO

# Condição: Exemplo

```
1. INÍCIO_ALGORITMO
2. DECLARE m
3. LEIA m
4. SE m >= 9
5. ENTÃO INÍCIO
6.     ESCREVA "aprovado summa cum laude"
7.     FIM
8. SENÃO INÍCIO
9.     SE m >= 8
10.    ENTÃO INÍCIO
11.        ESCREVA "aprovado cum laude"
12.        FIM
13.    SENÃO INÍCIO
14.        SE m >= 6
15.        ENTÃO INÍCIO
16.            ESCREVA "aprovado"
17.            FIM
18.        SENÃO INÍCIO
19.            ESCREVA "reprovado"
20.            FIM
21.        FIM
22.    FIM
23. FIM_ALGORITMO
```

BLOCO 1

BLOCO 2

# Condição: Exemplo

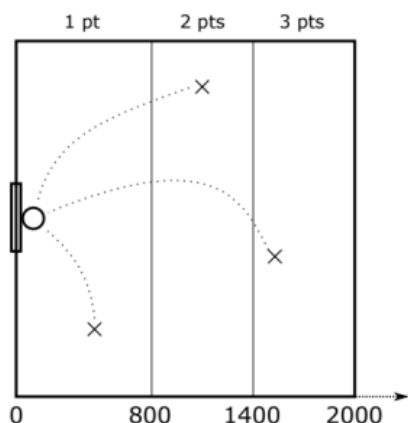
```
1. INÍCIO_ALGORITMO
2. DECLARE m
3. LEIA m
4. SE m >= 9
5. ENTÃO INÍCIO
6.     ESCREVA "aprovado summa cum laude"
7.     FIM
8. SENÃO INÍCIO
9.     SE m >= 8
10.    ENTÃO INÍCIO
11.        ESCREVA "aprovado cum laude"
12.        FIM
13.    SENÃO INÍCIO
14.        SE m >= 6
15.        ENTÃO INÍCIO
16.            ESCREVA "aprovado"
17.            FIM
18.        SENÃO INÍCIO
19.            ESCREVA "reprovado"
20.            FIM
21.        FIM
22.    FIM
23. FIM_ALGORITMO
```

BLOCO 1

BLOCO 2

# É possível diminuir o número de IFs dentro?

- SIM, usando a estrutura SENÃO SE



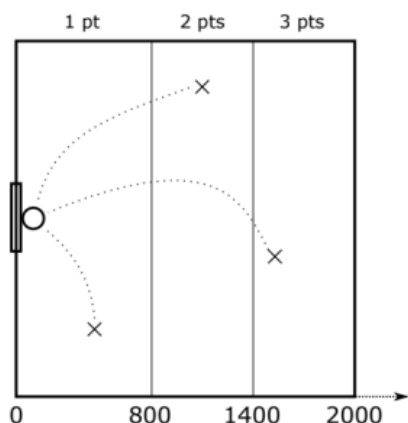
```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.         ESCREVA "1 ponto"          bloco1
7.     FIM
8. SENÃO SE d <= 1400
9. ENTÃO INÍCIO
10.        ESCREVA "2 pontos"         bloco2
11.    FIM
12. SENÃO SE d <= 2000
13. ENTÃO INÍCIO
14.        ESCREVA "3 pontos"         bloco3
15.    FIM
16. FIM_ALGORITMO
  
```

Condição Sendo Avaliada

# É possível diminuir o número de IFs dentro?

- SIM, usando a estrutura SENÃO SE



```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.         ESCREVA "1 ponto"
7.     FIM
8. SENÃO SE d <= 1400
9. ENTÃO INÍCIO
10.        ESCREVA "2 pontos"
11.    FIM
12. SENÃO SE d <= 2000
13. ENTÃO INÍCIO
14.        ESCREVA "3 pontos"
15.    FIM
16. FIM_ALGORITMO
  
```

Condição Sendo Avaliada

bloco1

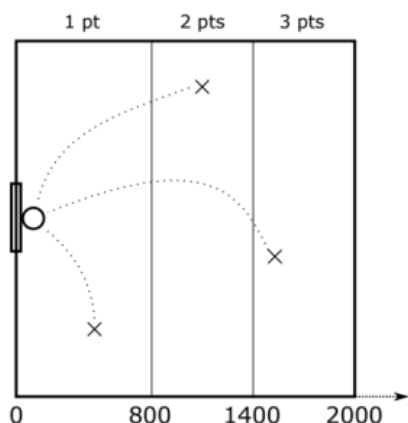
bloco2

bloco3



# É possível diminuir o número de IFs dentro?

- SIM, usando a estrutura SENÃO SE



```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.         ESCREVA "1 ponto"
7.     FIM
8. SENÃO SE d <= 1400
9. ENTÃO INÍCIO
10.        ESCREVA "2 pontos"
11.    FIM
12. SENÃO SE d <= 2000
13. ENTÃO INÍCIO
14.        ESCREVA "3 pontos"
15.    FIM
16. FIM_ALGORITMO
  
```

Condição Sendo Avaliada

bloco1

bloco2

bloco3





```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.     ESCREVA "1 ponto"      bloco1
7.     FIM
8. SENÃO SE d <= 1400
9. ENTÃO INÍCIO
10.     ESCREVA "2 pontos"    bloco2
11.     FIM
12. SENÃO SE d <= 2000
13. ENTÃO INÍCIO
14.     ESCREVA "3 pontos"    bloco3
15.     FIM
16. FIM_ALGORITMO

```

```

1. #include <stdio.h>
2. int main() {
3.     int d;
4.     scanf("%i", &d);
5.     if (d <= 800) {
6.         printf("1 ponto"); bloco1
7.     } else if (d <= 1400) {
8.         printf("2 pontos"); bloco2
9.     } else if (d <= 2000) {
10.        printf("3 pontos"); bloco3
11.    }
12. }

```



```

1. INÍCIO_ALGORITMO
2. DECLARE d
3. LEIA d
4. SE d <= 800
5. ENTÃO INÍCIO
6.     ESCREVA "1 ponto"      bloco1
7.     FIM
8. SENÃO SE d <= 1400
9. ENTÃO INÍCIO
10.     ESCREVA "2 pontos"    bloco2
11.     FIM
12. SENÃO INÍCIO
13.     ESCREVA "3 pontos"    bloco3
14.     FIM
15. FIM_ALGORITMO

```

```

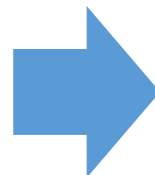
1. #include <stdio.h>
2. int main() {
3.     int d;
4.     scanf("%i", &d);
5.     if (d <= 800) {
6.         printf("1 ponto"); bloco1
7.     } else if (d <= 1400) {
8.         printf("2 pontos"); bloco2
9.     } else {
10.        printf("3 pontos"); bloco3
11.    }
12. }

```



■ senão se **condição** = else if (**condition**)

1. se  $n > 0$  então
2. escreva “positivo” bloco1
3. senão se  $n < 0$  então
4. escreva negativo bloco2
5. senão
6. escreva nulo bloco3
7. fimse



1. if ( $n > 0$ ) {
2. printf(“positivo”); bloco1
3. } else if ( $n < 0$ ) {
4. printf(“negativo”); bloco2
5. } else {
6. printf(“nulo”); bloco3
7. }




















## Else if e Else

- Else sem condição (if) captura todos os casos contrários
- Exemplo,  $d = 20$
- No caso, será testado as condições 4, 8, 12. Como todas são falsas, executa-se o senão (16)!





```
1.  INÍCIO_ALGORITMO
2.  DECLARE d
3.  LEIA d
4.  SE d == 0
5.  ENTÃO INÍCIO
6.      ESCREVA "nulo"
7.  FIM
8.  SENÃO SE d <= 10
9.  ENTÃO INÍCIO
10.     ESCREVA "menor que 10"
11.  FIM
12. SENÃO SE d <= 15
13.     INÍCIO
14.     ESCREVA "menor que 15"
15.  FIM
16. SENÃO INÍCIO
17.     ESCREVA "maior que 15"
18.  FIM
19. FIM_ALGORITMO
```








<div></div>			PRAÇAS DE PEDÁGIO							
			RODOVIA PRESIDENTE DUTRA							
CAT	Veículo		Imagem	Arujá (SP) km 204 Bidirecional	Guararema (SP) Km 182 Unidirecional	Guararema (SP) Km 180 Unidirecional	Jacarei (SP) Km 165 Bidirecional	Moreira César (SP) Km 87 Bidirecional	Itatiaia (RJ) Km 317 Bidirecional	Viúva Graça (RJ) Km 207 Bidirecional
1	<ul style="list-style-type: none"><li>Automóvel</li><li>Caminhonete</li><li>Furgão</li></ul>	Eixos 2		R\$ 3,70	R\$ 3,70	R\$ 3,70	R\$ 6,70	R\$ 15,20	R\$ 15,20	R\$ 15,20
		Rodagem 								
2	<ul style="list-style-type: none"><li>Caminhão Leve</li><li>Furgão</li><li>Ônibus</li><li>Caminhão - trator</li><li>Caminonete c/ 1 eixo duplo</li></ul>	Eixos 2		R\$ 7,40	R\$ 7,40	R\$ 7,40	R\$ 13,40	R\$ 30,40	R\$ 30,40	R\$ 30,40
		Rodagem 								
3	<ul style="list-style-type: none"><li>Caminhão/Ônibus</li><li>Caminhão - trator</li><li>Caminhão - trator com semi reboque</li><li>Caminhão/Ônibus 2 eixos c/ reboque de 1 eixo simples</li></ul>	Eixos 3		R\$ 11,10	R\$ 11,10	R\$ 11,10	R\$ 20,10	R\$ 45,60	R\$ 45,60	R\$ 45,60
		Rodagem 								
4	<ul style="list-style-type: none"><li>Caminhão com reboque</li><li>Caminhão - trator com semi reboque</li><li>Caminhão/Ônibus 2 eixos c/ reboque de 2 eixos simples</li></ul>	Eixos 4		R\$ 14,80	R\$ 14,80	R\$ 14,80	R\$ 26,80	R\$ 60,80	R\$ 60,80	R\$ 60,80
		Rodagem 								
5	<ul style="list-style-type: none"><li>Caminhão com reboque</li><li>Caminhão - trator com semi reboque</li></ul>	Eixos 5		R\$ 18,50	R\$ 18,50	R\$ 18,50	R\$ 33,50	R\$ 76,00	R\$ 76,00	R\$ 76,00
		Rodagem 								
6	<ul style="list-style-type: none"><li>Caminhão com reboque</li><li>Caminhão - trator com semi reboque</li></ul>	Eixos 6		R\$ 22,20	R\$ 22,20	R\$ 22,20	R\$ 40,20	R\$ 91,20	R\$ 91,20	R\$ 91,20
		Rodagem 								
7	<ul style="list-style-type: none"><li>Automóvel com semi reboque</li><li>Caminhonete com semi reboque</li></ul>	Eixos 3		R\$ 5,55	R\$ 5,55	R\$ 5,55	R\$ 10,05	R\$ 22,80	R\$ 22,80	R\$ 22,80
		Rodagem 								
8	<ul style="list-style-type: none"><li>Automóvel com reboque</li><li>Caminhonete com reboque</li></ul>	Eixos 4		R\$ 7,40	R\$ 7,40	R\$ 7,40	R\$ 13,40	R\$ 30,40	R\$ 30,40	R\$ 30,40
		Rodagem 								



# Preço do pedágio / depende da categoria

 <b>CCR NovaDutra</b>					
CAT	Veículo		Imagem	Arujá (SP) km 204 Bidirecional	
1	<ul style="list-style-type: none"> <li>Automóvel</li> <li>Caminhonete</li> <li>Furgão</li> </ul>	Eixos 2		R\$	3,70
		Rodagem			
2	<ul style="list-style-type: none"> <li>Caminhão Leve</li> <li>Furgão</li> <li>Ônibus</li> <li>Caminhão - trator</li> <li>Camionete c/ 1 eixo duplo</li> </ul>	Eixos 2		R\$	7,40
		Rodagem			
3	<ul style="list-style-type: none"> <li>Caminhão/Ônibus</li> <li>Caminhão - trator</li> <li>Caminhão - trator com semi reboque</li> <li>Caminhão/Ônibus 2 eixos c/ reboque de 1 eixo simples</li> </ul>	Eixos 3		R\$	11,10
		Rodagem			
	• Caminhão	Eixos			

4	<ul style="list-style-type: none"> <li>Caminhão com reboque</li> <li>Caminhão - trator com semi reboque</li> <li>Caminhão/Ônibus 2 eixos c/ reboque de 2 eixos simples</li> </ul>	Eixos 4		R\$	14,80
		Rodagem			
5	<ul style="list-style-type: none"> <li>Caminhão com reboque</li> <li>Caminhão - trator com semi reboque</li> </ul>	Eixos 5		R\$	18,50
		Rodagem			
6	<ul style="list-style-type: none"> <li>Caminhão com reboque</li> <li>Caminhão - trator com semi reboque</li> </ul>	Eixos 6		R\$	22,20
		Rodagem			
7	<ul style="list-style-type: none"> <li>Automóvel com semi reboque</li> <li>Caminhonete com semi reboque</li> </ul>	Eixos 3		R\$	5,55
		Rodagem			
8	<ul style="list-style-type: none"> <li>Automóvel com reboque</li> <li>Caminhonete com reboque</li> </ul>	Eixos 4		R\$	7,40
		Rodagem			





# Instrução SWITCH

- Útil quando queremos testar várias possibilidades de fluxo de código mas não queremos usar vários “*else if*”.
- Desta forma, cada possibilidade é testada em um bloco.

```
1. int exp = 2;  
2. switch (exp) {  
3.     case 0:  
4.         printf("igual a zero");  
5.         break;  
6.     case 3:  
7.         printf("igual a três");  
8.         break;  
9.     default:  
10.        printf("nenhum dos casos");  
11.}
```



# Instrução SWITCH

```
1. int exp = 2;
2. switch (exp) {                               avalie o valor de exp1
3.     case 0:                                   caso seja igual a 0
4.         printf("igual a zero");
5.         break;
6.     case 10:                                  caso seja igual a 3
7.         printf("igual a três");
8.         break;
9.     default:                                  caso não se encaixe em nenhum caso
10.        printf("nenhum dos casos");
11. }
```





```
switch(x) {  
    case 5:  
        // caso seja igual a 5  
        break;  
    case 0 ... 4:  
        // caso seja entre 0 e 4  
        break;  
    default:  
        // caso não se encaixe em nenhuma das categorias  
}
```



```
1. #include <stdio.h>
2.
3. int main() {
4.     int cat;
5.     printf("Digite a categoria: ");
6.     scanf("%i", &cat);
7.
8.     if (cat == 1) {
9.         printf("3.40");
10.    } else if (cat == 2) {
11.        printf("6.80");
12.    } else if (cat == 3) {
13.        printf("10.20");
14.    } else if (cat == 4) {
15.        printf("13.60");
16.    } else if (cat == 5) {
17.        printf("17.00");
18.    } else if (cat == 6) {
19.        printf("20.40");
20.    } else if (cat == 7) {
21.        printf("5.10");
22.    } else {
23.        printf("6.80");
24.    }
25.    return 0;
26.}
```

```
1. #include <stdio.h>
2.
3. int main() {
4.     int cat;
5.     printf("Digite a categoria: ");
6.     scanf("%i", &cat);
7.
8.     switch (cat) {
9.         case 1:
10.            printf("5.20");
11.            break;
12.         case 2:
13.            printf("6.80");
14.            break;
15.         case 3:
16.            printf("10.20");
17.            break;
18.         case 4:
19.            printf("13.60");
20.            break;
21.         case 5:
22.            printf("17.00");
23.            break;
24.         case 6:
25.            printf("20.40");
26.            break;
27.         case 7:
28.            printf("5.10");
29.            break;
30.         default:
31.            printf("6.80");
32.            break;
33.     }
34.     return 0;
35.}
```

# Exercício

<https://www.beecrowd.com.br/judge/pt/problems/view/1048>

- A empresa ABC resolveu conceder um aumento de salários a seus funcionários de acordo com a tabela abaixo.
- Leia o salário do funcionário e calcule e mostre o novo salário, bem como o valor de reajuste ganho e o índice reajustado, em percentual.

Salário	Percentual de Reajuste
0 - 400.00	15%
400.01 - 800.00	12%
800.01 - 1200.00	10%
1200.01 - 2000.00	7%
Acima de 2000.00	4%



# Exercício

<https://www.beecrowd.com.br/judge/pt/problems/view/1048>

- Entrada: A entrada contém apenas um valor de ponto flutuante, com duas casas decimais.
- Saída: Imprima 3 linhas na saída: o novo salário, o valor ganho de reajuste (ambos devem ser apresentados com 2 casas decimais) e o percentual de reajuste ganho, conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
400.00	Novo salario: 460.00 Reajuste ganho: 60.00 Em percentual: 15 %
800.01	Novo salario: 880.01 Reajuste ganho: 80.00 Em percentual: 10 %



# Exercício (Intervalo)

<https://www.beecrowd.com.br/judge/pt/problems/view/1037>

Você deve fazer um programa que leia um valor qualquer e apresente uma mensagem dizendo em qual dos seguintes intervalos

$[0,25]$ ,  $(25,50]$ ,  $(50,75]$ ,  $(75,100]$

este valor se encontra. Obviamente se o valor não estiver em nenhum destes intervalos, deverá ser impressa a mensagem “Fora de intervalo”.

O símbolo  $($  representa "maior que". Por exemplo:  $[0,25]$  indica valores entre 0 e 25.0000, inclusive eles.  $(25,50]$  indica valores maiores que 25 Ex: 25.00001 até o valor 50.0000000



# Exercício (Intervalo)

<https://www.beecrowd.com.br/judge/pt/problems/view/1037>

Entrada: Um número com ponto flutuante qualquer.

Saída: A saída deve ser uma mensagem conforme exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
25.01	Intervalo (25,50]
25.00	Intervalo [0,25]
100.00	Intervalo (75,100]



# Desafio

- <https://www.beecrowd.com.br/judge/pt/problems/view/1847>



# Obrigado!

## Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE  
TRANSPORTES**

**FCT**  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

Fonte dos ícones utilizados: flaticons.com





# Obrigado!

## Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE  
TRANSPORTES**

**FCT**  
FACULDADE DE  
CIÊNCIAS E TECNOLOGIA



**UFG**  
UNIVERSIDADE  
FEDERAL DE GOIÁS

Fonte dos ícones utilizados: flaticons.com

