

## Problemas resolvidos – comando if (desvio condicional)

Este documento apresenta uma série de exercícios resolvidos que exemplificam a utilização dos comandos de desvio condicional em diversas situações práticas. Digite os exemplos, execute e faça testes utilizando diferentes entradas.

**EXEMPLO 1** Escreva um programa que leia um número inteiro e determine se ele é par ou ímpar

### RESOLUÇÃO:

```
print('Digite um numero inteiro: ')
x = int(input())
if x % 2 == 0:
    print(x, 'é um numero PAR ')
else:
    print(x, 'é um numero IMPAR ')
```

**EXEMPLO 2** Escreva um programa que leia dois números inteiros e determine qual deles é o maior ou se eles são iguais.

### RESOLUÇÃO:

```
print('* * O MAIOR DENTRE DOIS NUMEROS * *')
print();

#solicita x e y
x = int(input('Digite um número inteiro: '))
y = int(input('Digite outro número inteiro: '))
print();

#identifica quem é o maior ou se eles são iguais
if x > y:
    print('o primeiro é maior ')
else:
    if x < y:
        print('o segundo é maior ')
    else:
        print('são iguais ')
```

O programa acima representa um exemplo de uso de comandos if “**aninhados**” ou “**encaixados**”. Isto significa a tomada de uma decisão dentro de outra decisão. No programa há um primeiro teste para determinar se “x” é maior do que “y”. Caso o resultado do teste seja verdadeiro, a mensagem ‘o primeiro é maior’ é impressa na tela. Senão, é preciso fazer outro teste (ou seja, um teste é feito dentro de outro teste!): se “x” for menor do que “y”, exibe-se ‘o segundo é maior’, senão exibe-se ‘são iguais’.

Para implementar uma rotina deste tipo – if dentro de else, if dentro de if, ou qualquer coisa parecida – também é preciso fazer uso da indentação.

**EXEMPLO 3.** Escrever um programa que funcione da seguinte maneira:

Inicialmente, o programa deve **ler** um **número inteiro** digitado pelo usuário e armazená-lo numa variável chamada  $x$ .

Em seguida, o sistema deve **calcular** e **imprimir** o valor da função  $y = f(x) + g(x)$ , onde:

$$h(x) = x^2 - 16$$

$$f(x) = \begin{cases} h(x), & \text{se } h(x) \geq 0 \\ 1, & \text{se } h(x) < 0 \end{cases}$$

$$g(x) = \begin{cases} x^2 + 16, & \text{se } f(x) = 0 \\ 0, & \text{se } f(x) > 0 \end{cases}$$

**RESOLUÇÃO:**

```
x = int(input('digite o valor de x (número inteiro): '))

#calcula h(x)
hx = x ** 2 - 16

#calcula f(x)
if hx >= 0:
    fx = hx
else:
    fx = 1

#calcula g(x)
if fx == 0:
    gx = x ** 2 + 16
else:
    gx = 0

#agora sim, posso calcular e imprimir y
y = fx + gx
print('y = ', y)
```

**EXEMPLO 4.** Numa fábrica, trabalham operários divididos em duas classes:

**A** – Os que fazem até 50 peças por mês.      **B** – Os que fazem mais de 50 peças por mês.

Os operários da classe A recebem R\$ 1500 por mês.

Já os empregados da classe B recebem R\$ 1500,00 mensais mais R\$ 25 por peça adicionalmente produzida (ou seja, peça além das 50 iniciais).

Escreva um programa que leia a classe do funcionário e o número de peças produzidas por ele no mês e calcule o seu salário.

```
#PASSO 1: RECEBE A CLASSE DO FUNCIONÁRIO
print('Digite a classe do funcionário:')
classe = input()

#PASSO 2: DETERMINA E IMPRIME O SALÁRIO DO FUNCIONÁRIO
if (classe == 'A') or (classe == 'a'):
```

```

    print('O salário do funcionário é R$ 1500.00')
else:
    if (classe == 'B') or (classe == 'b'):
        print('Entre com o número de peças fabricadas: ')
        pecas = int(input())
        if (pecas > 50):
            salario = 1500 + ((pecas - 50) * 25)
        else:
            salario = 1500
        print('O salário do funcionário eh R$ ', round(salario,2))
    else:
        print('VOCÊ DIGITOU UMA CLASSE DE FUNCIONARIO INEXISTENTE! ')

```

### Observações:

1. Para dar mais liberdade ao usuário, deixamos ele teclar as letras em minúsculo ou maiúsculo. Dessa forma, é preciso testar `if (classe == 'A') or (classe == 'a')`, ou seja, o valor da variável “classe” em maiúsculo e minúsculo (as letras maiúsculas e minúsculas possuem códigos internos diferentes).
2. Veja que o programa é inteligente o suficiente para não fazer o cálculo quando o usuário entra com uma classe inexistente (classe que não seja do tipo A ou B).

**EXEMPLO 5.** Escreva um programa que leia o número de um mês (ex.: 1 = Janeiro... 2 = Fevereiro) e o número de um ano (ex: 1970, 2002, 1958, etc) e imprima a quantidade de dias do mês.

**OBS:** é preciso considerar que o número de dias do mês de fevereiro é igual a 29 em anos bissextos. Para ser bissexto, o ano deve ser:

- múltiplo de 4, mas não pode ser múltiplo 100;
- ou é múltiplo de 400.

A resolução é apresentada na página a seguir. Analise com calma e teste o programa com diferentes entradas para facilitar o seu entendimento.

```

#PASSO 1: recebe o ano e mês
print('Digite o ano: ')
ano = int(input())

print('Digite o número do mês (1 a 12): ')
mes = int(input())

#PASSO 2: calcula e exibe a quantidade de dias
if mes > 12 or mes < 1:
    print('DADOS INVÁLIDOS!')
else:
    if (mes == 4) or (mes == 6) or (mes == 9) or (mes == 11):
        print('O MÊS TEM 30 DIAS')
    else:
        if (mes != 2):
            print('O MÊS TEM 31 DIAS')
        else: #se for fevereiro, verifica se é ano bissexto
            if (ano % 4 == 0 and ano % 100 != 0) or (ano % 400 == 0):
                print('O MÊS TEM 29 DIAS')
            else:
                print('O MÊS TEM 28 DIAS')

```

## **SOLUÇÃO 2 (com elif para evitar a indentação “chata”)**

*#PASSO 1: recebe o ano e mês*

```
print('Digite o ano: ')
```

```
ano = int(input())
```

```
print('Digite o número do mês (1 a 12): ')
```

```
mes = int(input())
```

*#PASSO 2: calcula e exibe a quantidade de dias*

```
if mes > 12 or mes < 1:
```

```
    print('DADOS INVÁLIDOS!')
```

```
elif (mes == 4) or (mes == 6) or (mes == 9) or (mes == 11):
```

```
    print('O MÊS TEM 30 DIAS')
```

```
elif (mes != 2):
```

```
    print('O MÊS TEM 31 DIAS')
```

```
else: #se for fevereiro, verifica se é ano bissexto
```

```
    if (ano % 4 == 0 and ano % 100 != 0) or (ano % 400 == 0):
```

```
        print('O MÊS TEM 29 DIAS')
```

```
    else:
```

```
        print('O MÊS TEM 28 DIAS')
```