

---

## Desafios 3 – Condicionais Encadeadas(if/elif)

---

### Questão 1

Escreva um programa que leia a **idade** e o **índice de massa corporal (IMC)** de uma pessoa, nessa ordem. Como saída, determine o risco de problemas cardíacos dessa pessoa, conforme a tabela a seguir:

		Idade	
		< 45	≥ 45
IMC	< 22,0	Baixo	Médio
	≥ 22,0	Médio	Alto

#### Validação dos dados

Verifique se os dados informados são válidos. Se a idade for menor ou igual a zero ou maior que 130 anos, ou se o IMC for menor ou igual a zero, imprima:

**Entradas: X anos e IMC Y**

**Dados invalidos**

Se as entradas forem **válidas**, imprima:

**Entradas: X anos e IMC Y**

**Risco: Z**

Nas mensagens de saída, substitua as letras **X**, **Y** e **Z** pelos valores fornecidos.

### Dicas

1. Use condições encadeadas (**if/elif**).
2. Atenção para o uso de caracteres maiúsculos e minúsculos. **Não use acentos.**
3. Alguns dos operadores booleanos (**and**, **or**, **not**) serão úteis.

## Exemplo de Entrada

25

21.0

## Exemplo de Saída

Entradas: 25 anos e IMC 21.0

Risco: Baixo

```
1 idade = int(input("1.Digite sua idade: "))
2 imc = float(input("2.Digite o valor de seu IMC(Indice de Massa Corporal): "))
3
4 print("Entradas:", idade, "anos e IMC", imc)
5
6 if (idade > 0) and (idade <= 130) and (imc > 0):
7     if (idade < 45) and (imc < 22.0):
8         print("Risco: Baixo")
9     elif ((idade < 45) and (imc >= 22)) or ((idade >= 45) and (imc < 22)):
10        print("Risco: Medio")
11    else:
12        print("Risco: Alto")
13 else:
14    print("Dados invalidos")
```

## Questão 2

Escreva um programa que leia o valor de **duas características** sobre **três imagens** de faces humanas. Como saída, determine quais as duas faces mais similares.

Uma das técnicas de comparação de faces usa como critério a razão ( $\rho$ ) entre duas medidas: a distância entre os olhos e a distância entre o nariz e o queixo. As duas imagens de face mais similares são aquelas cujas razões são mais próximas. Exemplo:

	Imagem 1	Imagem 2	Imagem 3
Distância entre os olhos	5,7	6,0	6,0
Distância entre o nariz e o queixo	5,3	5,0	5,6

- **Face 1:**  $\rho_1 = (5,7/5,3) = 1,08$
- **Face 2:**  $\rho_2 = (6,0/5,0) = 1,20$
- **Face 3:**  $\rho_3 = (6,0/5,6) = 1,07$

Quando comparamos as diferenças entre cada par de razões, vemos que a diferença entre as faces 1 e 3 é a menor de todas:  $|1,08 - 1,07| = 0,01$  significa que as faces mais semelhantes são a 1 e a 3, segundo essa técnica.

Considere a seguinte ordem das entradas:

1. distância entre os olhos, da Imagem 1
2. distância entre o nariz e o queixo, da Imagem 1
3. distância entre os olhos, da Imagem 2
4. distância entre o nariz e o queixo, da Imagem 2
5. distância entre os olhos, da Imagem 3
6. distância entre o nariz e o queixo, da Imagem 3

A saída deve ser o número das faces mais semelhantes, em ordem numérica crescente.

## Dicas

1. Atenção para a ordem de leitura de valores.
2. Use o operador lógico (**and**) para a comparação entre as diferenças.
3. Use a função **abs()** para encontrar o valor absoluto (módulo) de um número.

## Exemplo de Entrada

5.7

5.3

6.0

5.0

6.0

5.6

## Exemplo de Saída

1 e 3

```
1 dist_eye1 = float(input("1.Digite o valor da distancia entre o olhos na Imagem 1: "))
2 dist_nose1 = float(input("2.Digite o valor da distancia entre o nariz e o queixo na Imagem 1: "))
3
4 dist_eye2 = float(input("1.Digite o valor da distancia entre o olhos na Imagem 2: "))
5 dist_nose2 = float(input("2.Digite o valor da distancia entre o nariz e o queixo na Imagem 2: "))
6
7 dist_eye3 = float(input("1.Digite o valor da distancia entre o olhos na Imagem 3: "))
8 dist_nose3 = float(input("2.Digite o valor da distancia entre o nariz e o queixo na Imagem 3: "))
9
10 r1 = (dist_eye1) / (dist_nose1)
11 r2 = (dist_eye2) / (dist_nose2)
12 r3 = (dist_eye3) / (dist_nose3)
13
14 d1_2 = abs(r1 - r2)
15 d1_3 = abs(r1 - r3)
16 d2_3 = abs(r2 - r3)
17
18 if (min(d1_2, d1_3, d2_3)) == d1_2:
19     print("1 e 2")
20 elif (min(d1_2, d1_3, d2_3)) == d1_3:
21     print("1 e 3")
22 else:
23     print("2 e 3")
```

### Questão 3

Para participar de um jogo de loteria, uma pessoa deve apostar em um número com dois dígitos. Os prêmios são definidos de acordo com as seguintes regras:

Regra	Prêmio
O número que o jogador apostou é igual ao número sorteado pela loteria	R\$ 100.000,00
O número que o jogador apostou contém os mesmos dígitos do número sorteado na loteria, mas em ordem diferente	R\$ 50.000,00
Um dos dígitos do número que o jogador apostou é igual a um dos dígitos do número sorteado na loteria	R\$ 1.000,00

Escreva um programa que leia as seguintes informações:

- **Número** apostado pelo jogador (com dois dígitos)
- **Número** sorteado na loteria (também com dois dígitos)

O programa deverá verificar se o jogador ganhou ou perdeu.

- Se ganhar o primeiro prêmio, imprima: **Ganhou R\$ 100.000,00**
- Se ganhar o segundo prêmio, imprima: **Ganhou R\$ 50.000,00**
- Se ganhar o terceiro prêmio, imprima: **Ganhou R\$ 1.000,00**
- Caso contrário, imprima: **Perdeu**

### Dicas

1. Atenção para a ordem de leitura de valores.
2. Use os operadores booleanos (**and**, **or**).
3. Use os operadores de divisão inteira (**//**) e resto de divisão (**%**) para identificar os dígitos.
4. Assuma que o jogador digitou números entre 10 e 99.

### Exemplo de Entrada

23

32

### Exemplo de Saída

Ganhou R\$ 50.000,00

```
1 n_bet = int(input("1.Digite o numero(com dois digitos) apostado pelo jogador: "))
2 n_win = int(input("2.Digite o numero(com dois digitos) sorteado na loteria: "))
3
4 db1 = n_bet // 10
5 db2 = n_bet % 10
6
7 dw1 = n_win // 10
8 dw2 = n_win % 10
9
10 if (n_bet == n_win):
11 |     print("Ganhou R$ 100.000,00")
12 elif (db1 == dw2) and (db2 == dw1):
13 |     print("Ganhou R$ 50.000,00")
14 elif (db1 == dw1) or (db1 == dw2) or (db2 == dw1) or (db2 == dw2):
15 |     print("Ganhou R$ 1.000,00")
16 else:
17 |     print("Perdeu")
```

## Questão 4

A sensação de frio é determinada não só pela temperatura, mas também pela velocidade do vento, umidade relativa do ar e brilho do sol. Uma das formas de calcular a sensação de frio  $\sigma$  envolve a seguinte fórmula:

$$\sigma = 13,12 + 0,6215.T_{ar} - (11,37.v^{0,16}) + (0,3965.T_{ar}.v^{0,16}),$$

onde:

- $\sigma$  representa a sensação de frio
- $v$  representa a velocidade de vento em km/h
- $T_{ar}$  representa a temperatura do ar em graus Celsius (C)

Essa fórmula é definida apenas para temperaturas entre -50°C e 10°C, e velocidade acima de 4,8 km/h.

Escreva um programa que leia a temperatura em Celsius e a velocidade do vento em km/h.

- Se os valores de entrada forem válidos, o programa deverá calcular e exibir o valor de  $\sigma$ , com até **quatro casas** decimais de precisão.
- Se o valor da temperatura do ar for inválido, imprima: **Temperatura invalida**
- Se o valor da velocidade for inválido, imprima: **Velocidade invalida**

## Dicas

1. Atenção para a ordem de leitura de valores.
2. Use os operadores booleanos (**and**, **or**).
3. Use o comando **round(x, n)** para arredondar um valor **x** com até **n** casas decimais.

## Exemplo de Entrada

-5.3

9

## Exemplo de Saída

-9.3206

```
1 temp = float(input("1.Digite o valor da temperatura do ar em graus Celsius(C): "))
2 vent = float(input("2.Digite o valor da velocidade do vento em km/h: "))
3
4 if (temp > -50) and (temp < 10):
5     if (vent > 4.8):
6         sigma = 13.12 + 0.6215*temp - (11.37*vent**0.16) + (0.3965*temp*vent**0.16)
7         print(round(sigma, 4))
8     else:
9         print("Velocidade invalida")
10 else:
11     print("Temperatura invalida")
12     if (vent <= 4.8):
13         print("Velocidade invalida")
```



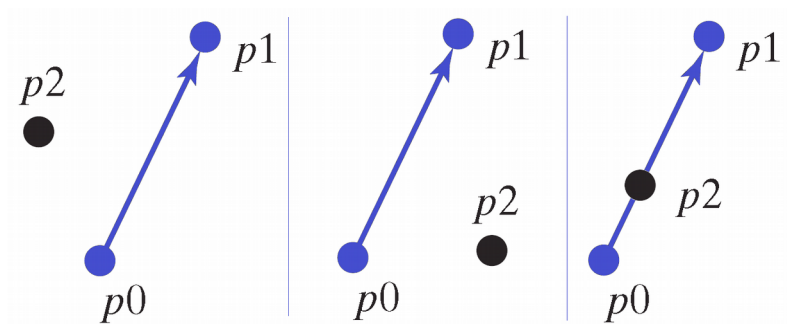
## Questão 5

Considere uma reta que parte do ponto  $p_0(x_0, y_0)$  ao ponto  $p_1(x_1, y_1)$ . Dado um ponto  $p_2(x_2, y_2)$ , Pode-se decidir se  $p_2$  está à esquerda, à direita ou sobre a reta (veja a figura), com base na seguinte condição  $c$ :

$$c = (x_1 - x_0) \cdot (y_2 - y_0) - (x_2 - x_0) \cdot (y_1 - y_0)$$

A localização de  $p_2$  é definida pelas seguintes regras:

- Se  $c < 0$ ,  $p_2$  está à **direita** da reta.
- Se  $c > 0$ ,  $p_2$  está à **esquerda** da reta.
- Se  $c = 0$ ,  $p_2$  está **sobre** a reta.



Escreva um programa que leia as **coordenadas**  $x$  e  $y$  de cada um dos três pontos  $p_0, p_1$  e  $p_2$ , um par  $(x_i, y_i)$  por vez. O programa deverá calcular o valor de  $c$  e imprimir a localização de  $p_2$ :

- A direita da reta
- A esquerda da reta
- Sobre a reta

Fonte: Liang. Introduction to Programming Using Python. Pearson, 2013.

## Dicas

1. Atenção para a ordem de leitura de valores.
2. Não se limite ao caso de exemplo. Teste as diversas possibilidades de entradas e saídas.

## Exemplo de Entrada

1.0

1.0

5.0

5.0

2.0

2.0

## Exemplo de Saída

Sobre a reta

```
1 x0 = float(input("1.Digite o valor da coordenada X do ponto 0: "))
2 y0 = float(input("2.Digite o valor da coordenada Y do ponto 0: "))
3
4 x1 = float(input("1.Digite o valor da coordenada X do ponto 1: "))
5 y1 = float(input("2.Digite o valor da coordenada Y do ponto 1: "))
6
7 x2 = float(input("1.Digite o valor da coordenada X do ponto 2: "))
8 y2 = float(input("2.Digite o valor da coordenada Y do ponto 2: "))
9
10 c = (x1 - x0) * (y2 - y0) - (x2 - x0) * (y1 - y0)
11
12 if (c < 0):
13     print("A direita da reta")
14 if (c > 0):
15     print("A esquerda da reta")
16 if (c == 0):
17     print("Sobre a reta")
```

## Questão 6

Em uma empresa, o salário líquido dos funcionários é calculado em função de dois descontos: a contribuição previdenciária e o imposto de renda. Dado o salário bruto do funcionário, a contribuição previdenciária é calculado de acordo com a seguinte tabela:

Salário bruto	Alíquota
Até R\$ 1.659,38	8%
De R\$ 1.659,39 a R\$ 2.765,66	9%
De R\$ 2.765,67 a R\$ 5.531,31	11%
Acima de R\$ 5.531,31	Desconto fixo de R\$ 608,44

**Após o desconto da contribuição previdenciária**, o valor restante é então utilizado para calcular o desconto do imposto de renda, com base nesta tabela:

Salário base	Alíquota de desconto
Até R\$ 1.903,98	Isento
De R\$ 1.903,99 a R\$ 2.826,65	7,5%
De R\$ 2.826,66 a R\$ 3.751,05	15%
De R\$ 3.751,06 a R\$ 4.664,68	22,5%
Acima de R\$ 4.664,68	27,5%

Escreva um programa que leia o **salário bruto** de uma pessoa, e aplique os dois descontos. Como saída, imprima a seguinte:

**Salario liquido = R\$ X**

### Dicas

1. Primeiro calcule a contribuição previdenciária e desconte do salário bruto. Depois, use o valor restante para determinar ao desconto do imposto de renda. Por fim, subtraia esse valor e determine o valor do salário líquido.
2. Use o comando **round(x, n)** para arredondar a resposta **x** com até **n** casas decimais.
3. Use o operador lógico (**and**) para a comparação entre as diferenças.

### Exemplo de Entrada

5000.00

### Exemplo de Saída

Salario liquido = R\$ 3448.75

```
1 total = float(input("1.Digite o respectivo valor do salario bruto: R$ "))
2
3 if (total <= 1659.38):
4     rest = 0.92*total
5 if (total >= 1659.39) and (total <= 2765.66):
6     rest = 0.91*total
7 if (total >= 2765.67) and (total <= 5531.31):
8     rest = 0.89*total
9 if (total > 5531.31):
10     rest = total - 608.44
11
12 if (rest >= 1903.99) and (rest <= 2826.65):
13     rest = 0.925*rest
14 if (rest >= 2826.66) and (rest <= 3751.05):
15     rest = 0.85*rest
16 if (rest >= 3751.06) and (rest <= 4664.68):
17     rest = 0.775*rest
18 if (rest > 4664.68):
19     rest = 0.725*rest
20
21 print("Salario liquido = R$", round(rest,2))
```

## Questão 7

Uma prova é composta por três questões de múltipla escolha (a, b, c, d, e).

Escreva um programa que leia as **respostas** dadas por um estudante às três questões. Em seguida, leia o **gabarito** esperado para essas mesmas três questões.

Como saída, indique **quantas** respostas estão corretas.

## Dicas

1. Compare cada questão em um **if** independente. Não é necessário usar **elif**.
2. Crie uma variável de contagem com valor inicial igual a zero. Some 1 a ela sempre que a resposta do aluno conferir com o gabarito esperado.
3. Lembre-se que **"x"** (com aspas) indica o caractere xis, ao passo que **x** (sem aspas) indica o nome de uma variável chamada **x**.

## Exemplo de Entrada

a  
b  
c  
e  
d  
c

## Exemplo de Saída

1

```
1 q1 = input("1.Digite a resposta dada pelo estudante na Questao 1: ")
2 q2 = input("2.Digite a resposta dada pelo estudante na Questao 2: ")
3 q3 = input("3.Digite a resposta dada pelo estudante na Questao 3: ")
4
5 r1 = input("1.Digite a resposta encontrada no gabarito referente a Questao 1: ")
6 r2 = input("2.Digite a resposta encontrada no gabarito referente a Questao 2: ")
7 r3 = input("3.Digite a resposta encontrada no gabarito referente a Questao 3: ")
8
9 acertos = 0
10
11 if (q1 == r1):
12     acertos = acertos + 1
13 if (q2 == r2):
14     acertos = acertos + 1
15 if (q3 == r3):
16     acertos = acertos + 1
17
18 print(acertos)
```

## Questão 8

A professora de Introdução à Programação de Computadores resolveu dividir a turma em três grupos para realização de uma atividade, conforme a primeira letra do nome de cada estudante:

- A - K : sala 101
- L - N : sala 102
- O - Z : sala 103

Escreva um programa que leia o **nome** de um(a) estudante. Como saída, imprima a seguinte mensagem:

**Fulano(a) vai para a sala X**

Substitua **Fulano(a)** pelo nome fornecido na entrada e **X** pelo número da sala correspondente.

## Dicas

1. A primeira letra de uma string **nome** é denotada por **nome[0]**
2. Durante a comparação, lembre-se que o usuário pode ter digitado o nome do estudante com inicial minúscula. Portanto, aplique o método **.upper()** ao final da primeira letra durante a comparação.

## Exemplo de Entrada

Ada Lovelace

## Exemplo de Saída

Ada Lovelace vai para a sala 101

```
1 nome = input("1.Digite o nome do aluno: ")
2
3 if (nome[0].upper() == 'A') or (nome[0].upper() == 'B') or (nome[0].upper() == 'C'):
4     print(nome, "vai para a sala 101")
5 elif (nome[0].upper() == 'D') or (nome[0].upper() == 'E') or (nome[0].upper() == 'F'):
6     print(nome, "vai para a sala 101")
7 elif (nome[0].upper() == 'G'):
8     print(nome, "vai para a sala 101")
9 elif (nome[0].upper() == 'H') or (nome[0].upper() == 'I') or (nome[0].upper() == 'J'):
10    print(nome, "vai para a sala 101")
11 elif (nome[0].upper() == 'K'):
12    print(nome, "vai para a sala 101")
13 elif (nome[0].upper() == 'L') or (nome[0].upper() == 'M') or (nome[0].upper() == 'N'):
14    print(nome, "vai para a sala 102")
15 else:
16    print(nome, "vai para a sala 103")
```

## Questão 9

O preço da passagem de barco, partindo de Manaus, varia conforme o destino desejado e se o passageiro fecha ou não um pacote de ida-e-volta. Escreva um programa que leia as seguintes informações, nesta ordem:

1. O nome da cidade de **destino**.
2. Se o percurso é somente de ida ou ida-e-volta.

Como saída, imprima o preço da passagem conforme a tabela abaixo. Se a cidade de destino não estiver na relação abaixo, imprima: **Destino inexistente**

Destino	Ida (R\$)	Ida-e-volta (R\$)
Belém	350,00	650,00
Borba	80,00	152,00
Coari	106,00	199,00
Humaitá	200,00	390,00
Manicoré	150,00	280,00
Maués	100,00	190,00

## Dicas

1. Atenção para o tipo das entradas e saídas (inteiro, real, string).
2. Não utilize acentos.
3. Use o ponto para separar a parte inteira da fracionária em números reais.

## Exemplo de Entrada

Belem

ida-e-volta

## Exemplo de Saída

650.0

```
1 cidade = input("1.Digite o nome da cidade de destino: ")
2 p = input("ida-e-volta || ida || Escolha: ")
3
4 if (cidade == "Belem"):
5     if(p == "ida"):
6         print("350.00")
7     if(p == "ida-e-volta"):
8         print("650.00")
9 elif (cidade == "Borba"):
10     if(p == "ida"):
11         print("80.00")
12     if(p == "ida-e-volta"):
13         print("152.00")
14 elif (cidade == "Coari"):
15     if(p == "ida"):
16         print("106.00")
17     if(p == "ida-e-volta"):
18         print("199.00")
19 elif (cidade == "Humaita"):
20     if(p == "ida"):
21         print("200.00")
22     if(p == "ida-e-volta"):
23         print("390.00")
24 elif (cidade == "Manicore"):
25     if(p == "ida"):
26         print("150.00")
27     if(p == "ida-e-volta"):
28         print("280.00")
29 elif (cidade == "Maues"):
30     if(p == "ida"):
31         print("100.00")
32     if(p == "ida-e-volta"):
33         print("190.00")
34 else:
35     print("Destino inexistente")
```



## Questão 10

Uma fórmula para determinar a data da Páscoa depende das seguintes variáveis:

- $a = ano \% 19$
- $b = ano \% 4$
- $c = ano \% 7$
- $d = (19a + 24) \% 30$
- $e = (2b + 4c + 6d + 5) \% 7$

A data da Páscoa será igual a **22 de março** +  $d + e$ , que pode cair em abril.

Essa fórmula funciona para todos os anos no intervalo 1900-2099, inclusive, exceto 1954, 1981, 2049 e 2076. Nesses quatro anos, a fórmula produz uma data que é uma **semana depois** da data verdadeira.

Escreva um programa que leia um **ano** inserido pelo usuário.

- Caso não esteja no intervalo adequado, imprima: **ano invalido**
- Caso contrário, imprima a data da Páscoa naquele ano: **D - M - AAAA**

### Dicas

1. Use o operador de resto da divisão inteira (%).
2. Note que março tem 31 dias. Os dias excedentes caem em abril.
3. Para testar seu programa, consulte esta [tabela](#) contendo as datas da Páscoa de 1951 a 2078.

### Exemplo de Entrada

2018

### Exemplo de Saída

1 - 4 - 2018

```
1 ano = int(input("1.Digite o ano no qual deseja saber a data da Pascoa: "))
2
3 dia = 0
4
5 if (ano >= 1900) and (ano <= 2099):
6     a = ano%19
7     b = ano%4
8     c = ano%7
9     d = (19*a + 24)%30
10    e = (2*b + 4*c + 6*d +5)%7
11
12    dia = (22 + d + e)
13
14    if (ano != 1954) and (ano != 1981) and (ano != 2049) and (ano != 2076):
15        if (dia > 31):
16            dia = dia - 31
17            print(dia,"- 4 -", ano)
18        else:
19            print(dia,"- 3 -", ano)
20    else:
21        dia = dia - 7
22        if (dia > 31):
23            dia = dia - 31
24            print(dia,"- 4 -", ano)
25        else:
26            print(dia,"- 3 -", ano)
27 else:
28    print("ano invalido")
```

## Questão 11

A [congruência de Zeller](#) é um algoritmo desenvolvido por Christian Zeller (1822-1899) para calcular o dia da semana. A fórmula é dada por:

$$h = \left( q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + k + \left\lfloor \frac{k}{4} \right\rfloor + \left\lfloor \frac{j}{4} \right\rfloor + 5j \right) \% 7$$

onde:

- $h$  é o dia da semana (**0** = sábado, **1** = domingo, **2** = segunda-feira, **3** = terça-feira, **4** = quarta-feira, **5** = quinta-feira, **6** = sexta-feira).
- $q$  é o dia do mês.
- $m$  é o mês (**3** = março, **4** = abril, **5** = maio, ..., **12** = dezembro). Janeiro e fevereiro são contados como meses 13 e 14 do ano anterior.
- $k$  é o ano do século, ou seja,  $ano \% 100$ .
- $j$  é o século\*, isto é,  $\lfloor ano / 100 \rfloor$ . (ver dica 6)

Escreva um programa que leia três entradas:

1. **dia**( $q$ ) do mês: um inteiro de 1 a 31.
2. **mês**: um inteiro de 1 a 12.
3. **ano**: um inteiro  $\geq 1$ .

Como saída, imprima o **nome** do dia da semana.

## Dicas

1. O piso (floor) de um número real  $x$ , denotado por  $\lfloor x \rfloor$ , é o resultado do arredondamento de  $x$  para baixo. Por exemplo,  $\lfloor 3,99 \rfloor = 3$ . Em Python, o piso de  $x$  é dado por **int(x)**, desde que  $x \geq 0$ .
2. Considere que as entradas são **sempre** válidas.
3. Se a entrada for **2 / 2 / 2010**, o algoritmo de Zeller conta a data como sendo o 2º dia do mês 14 de 2009, ou seja: 2 / 14 / 2009.
  - Portanto, se o mês lido for 1 ou 2, você deve somá-lo com 12 para obter  $m$ , e subtrair 1 do ano lido.
4. A saída do seu programa dependerá do valor de  $h$ .
5. Imprima o nome dos dias da semana em letras **minúsculas, sem acentos e sem espaços**.
6. De forma mais precisa,  $j$  denota o século se a contagem dos anos e séculos começasse em zero. Por exemplo, para os anos 1995 e 2000, temos que  $j = 19$  e  $j = 20$ , respectivamente. Na contagem comum, porém, ambos são considerados como sendo do século 20.

## Exemplo de Entrada

4

1

1643

## Exemplo de Saída

domingo

```
1 dia = int(input("1.Digite o numero do DIA da data a ser analisada pelo algoritmo: "))
2 mes = int(input("2.Digite o numero do MES da data a ser analisada pelo algoritmo: "))
3 ano = int(input("3.Digite o ANO da data a ser analisade pelo algoritmo: "))
4
5 q = dia
6 m = mes
7
8 if (mes == 1):
9     ano=ano-1
10    m = 13
11 if (mes == 2):
12     ano=ano-1
13     m = 14
14
15 k = ano%100
16 j = int(ano/100)
17
18 h = (q + int((13 * (m + 1))/5) + k + int(k/4) + int(j/4) + 5*j)%7
19
20 if (h == 0):
21     print("sabado")
22 elif (h == 1):
23     print("domingo")
24 elif (h == 2):
25     print("segunda-feira")
26 elif (h == 3):
27     print("terca-feira")
28 elif (h == 4):
29     print("quarta-feira")
30 elif (h == 5):
31     print("quinta-feira")
32 else:
33     print("sexta-feira")
```

## Questão 12

Escreva um programa que leia uma data como um número inteiro no formato **DDMMAAAA**.

- Se a data for válida, imprima: **DD de MM de AAAA eh uma data valida**
- Caso contrário, imprima: **DD de MM de AAAA nao eh uma data valida**

Para uma data ser válida, deve obedecer às seguintes regras:

1. O ano deve ser maior do que zero, pois não existiu ano zero.
2. O mês deve ser um valor entre 1 e 12.
3. O dia deve ser maior ou igual a 1, e o valor máximo dependerá do mês.
4. Em anos bissextos, o dia 29 de fevereiro é válido, mas não nos demais anos.
5. Em 1582, os dias 5 a 14 de outubro (inclusive) não existiram, pois foram suprimidos pelo Papa Gregório XIII para um ajuste do calendário, que por isso passou a se chamar "gregoriano".

### Dicas

1. Utilize o operador de resto da divisão (%) para separar a entrada em dias, mês e ano.
2. Utilize operadores relacionais.

### Exemplo de Entrada

31041500

### Exemplo de Saída

31 de 4 de 1500 nao eh uma data valida

```

1 data = int(input("1.Digite uma data a ser analisada no formato(DDMMAAAA): "))
2
3 dia = data // 1000000
4 mes = ((data % 10000000) % 1000000) // 10000
5 ano = data % 10000
6
7 aux = str(dia) + " de " + str(mes) + " de " + str(ano)
8 b = 0
9
10 if (ano > 0) and ((mes >= 1) and (mes <= 12)) and (dia >= 1):
11     if (ano != 1582) or ((ano == 1582) and (mes == 10) and ((dia < 5) or (dia > 14))):
12         # calculo do ano bissexto
13         if (ano % 4 == 0) or ((ano % 4 == 0) and (ano % 100 == 0) and (ano % 400 == 0)):
14             b = 1
15
16         if (mes == 1) and (dia <= 31):
17             print(aux + " eh uma data valida")
18         elif ((mes == 2) and (b == 1) and (dia <= 29)) or ((mes == 2) and (b == 0) and (dia <= 28)):
19             print(aux + " eh uma data valida")
20         elif (mes == 3) and (dia <= 31):
21             print(aux + " eh uma data valida")
22         elif (mes == 4) and (dia <= 30):
23             print(aux + " eh uma data valida")
24         elif (mes == 5) and (dia <= 31):
25             print(aux + " eh uma data valida")
26         elif (mes == 6) and (dia <= 30):
27             print(aux + " eh uma data valida")
28         elif (mes == 7) and (dia <= 31):
29             print(aux + " eh uma data valida")
30         elif (mes == 8) and (dia <= 31):
31             print(aux + " eh uma data valida")
32         elif (mes == 9) and (dia <= 30):
33             print(aux + " eh uma data valida")
34         elif (mes == 10) and (dia <= 31):
35             print(aux + " eh uma data valida")
36         elif (mes == 11) and (dia <= 30):
37             print(aux + " eh uma data valida")
38         elif (mes == 12) and (dia <= 31):
39             print(aux + " eh uma data valida")
40         else:
41             print(aux + " nao eh uma data valida")
42     else:
43         print(aux + " nao eh uma data valida")
44 else:
45     print(aux + " nao eh uma data valida")

```