

MAC 115 – Introdução à Ciência da Computação

Aula 13

Nelson Lago

IF noturno – 2023



Previously on MAC 115...

Repetições encaixadas

- Repetições usam uma variável de controle

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**
 - ▶ De maneira unidimensional!

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**
 - ▶ De maneira unidimensional!
 - ▶ Números naturais, nomes de pessoas, notas de alunos...

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**
 - ▶ De maneira unidimensional!
 - ▶ Números naturais, nomes de pessoas, notas de alunos...
- **Como trabalhar com conjuntos de maneira bidimensional?**

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**
 - ▶ De maneira unidimensional!
 - ▶ Números naturais, nomes de pessoas, notas de alunos...
- **Como trabalhar com conjuntos de maneira bidimensional?**
 - ▶ Pontos de um plano, tabelas...

Repetições encaixadas

- **Repetições usam uma variável de controle**
- **Quando essa variável corresponde a um conjunto, as repetições permitem trabalhar sobre os elementos desse conjunto**
 - ▶ De maneira unidimensional!
 - ▶ Números naturais, nomes de pessoas, notas de alunos...
- **Como trabalhar com conjuntos de maneira bidimensional?**
 - ▶ Pontos de um plano, tabelas...
- **Repetições encaixadas**

Repetições encaixadas

```
while condição1:  
    while condição2:  
        ...
```

Repetições encaixadas

```
while condição1:  
    while condição2:  
        ...
```

- A cada “rodada” do laço mais externo, o laço interno é executado várias vezes (até sua condição deixar de ser verdadeira)

Repetições encaixadas

```
while condição1:  
    while condição2:  
        ...
```

- A cada “rodada” do laço mais externo, o laço interno é executado várias vezes (até sua condição deixar de ser verdadeira)
- Para isso funcionar, **condição2** deve voltar a ser verdadeira a cada nova rodada do laço mais externo

Exemplo – tabuada



Exemplo – tabuada

```
tabuada_do = 1
```

Exemplo – tabuada

```
tabuada_do = 1  
while tabuada_do <= 10:
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:

    tabuada_do += 1
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:

    while n <= 10:

        tabuada_do += 1
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:
    n = 1
    while n <= 10:

        tabuada_do += 1
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:
    n = 1
    while n <= 10:
        print(tabuada_do * n)

    tabuada_do += 1
```

Exemplo – tabuada

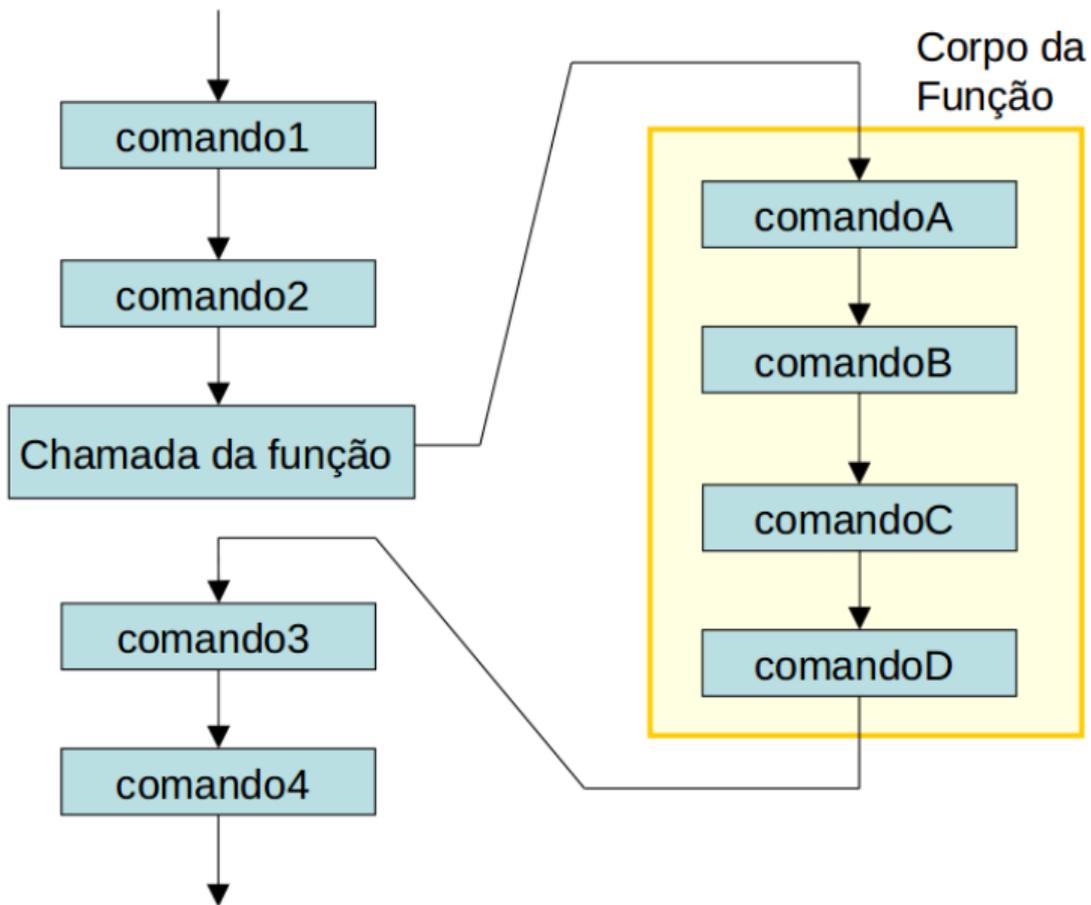
```
tabuada_do = 1
while tabuada_do <= 10:
    n = 1
    while n <= 10:
        print(tabuada_do * n)
        n += 1
    tabuada_do += 1
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:
    n = 1
    while n <= 10:
        print(tabuada_do * n , end="\t")
        n += 1
    tabuada_do += 1
```

Exemplo – tabuada

```
tabuada_do = 1
while tabuada_do <= 10:
    n = 1
    while n <= 10:
        print(tabuada_do * n , end="\t")
        n += 1
    tabuada_do += 1
    print()
```



- **Funções são inspiradas nas funções matemáticas**
- **Em geral, recebem parâmetros e devolvem valores (resultados) que dependem desses parâmetros**
 - ▶ Mas nem sempre! `print()`, por exemplo, não devolve nenhum resultado
 - » “Efeitos colaterais”

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
 - ▶ **print()** diz o que fazer com o valor de uma expressão

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
 - ▶ `print()` diz o que fazer com o valor de uma expressão
 - ▶ O operador = *também* diz o que fazer com o valor de uma expressão

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não é uma expressão*
 - ▶ `print()` diz o que fazer com o valor de uma expressão
 - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
 - ▶ `print()` diz o que fazer com o valor de uma expressão
 - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)
 - ▶ E, portanto, podem fazer parte de uma expressão

Nem tudo são expressões

Isto faz sentido?

```
x = print(2 + 3)
```

- **Nem tudo são expressões!**
- **2 + 3 é uma expressão**
- **print()** *não* é uma expressão
 - ▶ `print()` diz o que fazer com o valor de uma expressão
 - ▶ O operador `=` *também* diz o que fazer com o valor de uma expressão
- **int()** e **float()** são **quase** expressões (são *funções*)
 - ▶ E, portanto, podem fazer parte de uma expressão

```
x = 2 + int(3.7)
```

```
print(2 + int(3.7))
```

Funções – Efeitos colaterais

Sem efeito colateral:

```
def media(a, b):  
    return (a + b) / 2  
  
print(media(5, 7))
```

Com efeito colateral:

```
def media(a, b):  
    print((a + b) / 2)  
  
media(5, 7)
```

Funções – Efeitos colaterais

Sem efeito colateral:

```
def media(a, b):  
    return (a + b) / 2  
  
print(media(5, 7))
```

Com efeito colateral:

```
def media(a, b):  
    print((a + b) / 2)  
  
media(5, 7)
```

A segunda versão nem precisa ter **return**

Funções – Efeitos colaterais

Sem efeito colateral:

```
def media(a, b):  
    return (a + b) / 2  
  
print(media(5, 7))
```

Com efeito colateral:

```
def media(a, b):  
    print((a + b) / 2)  
  
media(5, 7)
```

A segunda versão nem precisa ter **return** (mas pode ter)

Funções – Efeitos colaterais

Sem efeito colateral:

```
def media(a, b):  
    return (a + b) / 2  
  
print(media(5, 7))
```

Com efeito colateral:

```
def media(a, b):  
    print((a + b) / 2)  
    return (a + b) / 2  
  
media(5, 7)
```

A segunda versão nem precisa ter **return** (mas pode ter)

Não precisa ter só um return:

```
def maximo(a, b):  
    if a > b:  
        return a  
    else:  
        return b  
  
print(maximo(2, 7))
```

Dado um número inteiro $n \geq 2$, diga se ele é primo

```
def éPrimo(x):  
    divisor = x - 1  
    primo = True  
    while divisor >= 2:  
        if x % divisor == 0:  
            primo = False  
        divisor -= 1  
    return primo
```

Dado um número inteiro $n \geq 2$, diga se ele é primo

```
def éPrimo(x):  
    divisor = x - 1  
    primo = True  
    while divisor >= 2:  
        if x % divisor == 0:  
            primo = False  
        divisor -= 1  
    return primo
```

- Não confunda **print()** e **return!**

Dado um número inteiro $n \geq 2$, diga se ele é primo

```
def éPrimo(x):  
    divisor = x - 1  
    primo = True  
    while divisor >= 2:  
        if x % divisor == 0:  
            primo = False  
        divisor -= 1  
    return primo
```

- Não confunda **print()** e **return!**

Funções – escopo

```
def fatorial(n):  
    fat = 1  
    while n >= 2:  
        fat *= n  
        n -= 1  
    return fat  
fatorial(4)  
print(fat)
```

Funções – escopo

```
def fatorial(n):  
    fat = 1  
    while n >= 2:  
        fat *= n  
        n -= 1  
    return fat  
print(fatorial(4))
```

Funções – main()

```
print(fatorial(4))  
def fatorial(n):  
    fat = 1  
    while n >= 2:  
        fat *= n  
        n -= 1  
    return fat
```

Funções – main()

```
def main():  
    x = int(input("Digite um inteiro positivo: "))  
    print(fatorial(x))
```

```
def fatorial(n):  
    fat = 1  
    while n >= 2:  
        fat *= n  
        n -= 1  
    return fat
```

```
main()
```

Funções – main()

```
def fatorial(n):  
    fat = 1  
    while n >= 2:  
        fat *= n  
        n -= 1  
    return fat  
  
def main():  
    x = int(input("Digite um inteiro positivo: "))  
    print(fatorial(x))  
  
main()
```

Embora não seja obrigatório, em geral é uma boa ideia usar uma função `main()`

Exercício – sistema de *login*

Imagine um sistema de *login* com três usuários:

- **Alan Turing** – UID **turing**, senha **tmachine**
- **Ada Lovelace** – UID **llace**, senha **anengine**
- **Grace Hopper** – UID **hopper**, senha **business**

Crie um sistema que lê o UID (*login*) e a senha do usuário e, se os dados estiverem corretos, escreve “Bem-vindo, [nome]!”; caso contrário, o sistema escreve “Login ou senha incorreto”.

Exercício – sistema de *login*



Exercício – sistema de *login*

```
uid = input("username: ")  
senha = input("senha: ")
```

Exercício – sistema de *login*

```
uid = input("username: ")
senha = input("senha: ")

if user_ok:
    print("Bem-vindo, {}".format(nome))
else:
    print("Login ou senha incorreto")
```

Exercício – sistema de *login*

```
uid = input("username: ")
senha = input("senha: ")
user_ok = False

if user_ok:
    print("Bem-vindo, {}".format(nome))
else:
    print("Login ou senha incorreto")
```

Exercício – sistema de *login*

```
uid = input("username: ")
senha = input("senha: ")
user_ok = False
if uid == "turing" and senha == "tmachine":
    user_ok = True
    nome = "Alan Turing"

if user_ok:
    print("Bem-vindo, {}".format(nome))
else:
    print("Login ou senha incorreto")
```

Exercício – sistema de *login*

```
uid = input("username: ")
senha = input("senha: ")
user_ok = False
if uid == "turing" and senha == "tmachine":
    user_ok = True
    nome = "Alan Turing"
elif uid == "llace" and senha == "anengine":
    user_ok = True
    nome = "Ada Lovelace"
elif uid == "hopper" and senha == "business":
    user_ok = True
    nome = "Grace Hopper"
if user_ok:
    print("Bem-vindo, {}".format(nome))
else:
    print("Login ou senha incorreto")
```

Exercício – sistema de *login*

```
def main():
    uid = input("username: ")
    senha = input("senha: ")
    nome = checa_login(uid, senha)
    if nome == "":
        print("Login ou senha incorreto")
    else:
        print("Bem-vindo, {}".format(nome))
```

Exercício — sistema de *login*

```
def main():
    uid = input("username: ")
    senha = input("senha: ")
    nome = checa_login(uid, senha)
    if nome == "":
        print("Login ou senha incorreto")
    else:
        print("Bem-vindo, {}".format(nome))

def checa_login(uid, senha):
    nome = ""
    if uid == "turing" and senha == "tmachine":
        nome = "Alan Turing"
    elif uid == "llace" and senha == "anengine":
        nome = "Ada Lovelace"
    elif uid == "hopper" and senha == "business":
        nome = "Grace Hopper"
    return nome
```

Exercício – sistema de *login*

Modifique o código anterior para que o programa faça saudações diferentes em função do horário:

- “Bom dia, [nome]!” (das 6h às 12h59)
- “Boa tarde, [nome]!” (das 13h às 18h59)
- “Boa noite, [nome]!” (após as 19h ou antes das 6h)

Para saber o horário atual:

```
import time
hora = int(time.strftime('%H'))
```

Exercício – sistema de *login*

```
def checa_login(uid, senha):
    nome = ""
    if uid == "turing" and senha == "tmachine":
        nome = "Alan Turing"
    elif uid == "llace" and senha == "anengine":
        nome = "Ada Lovelace"
    elif uid == "hopper" and senha == "business":
        nome = "Grace Hopper"
    return nome
```

Exercício – sistema de *login*

```
def main():
    uid = input("username: ")
    senha = input("senha: ")
    nome = checa_login(uid, senha)
    if nome == "":
        print("Login ou senha incorreto")
    else:
```

Exercício – sistema de *login*

```
def main():
    uid = input("username: ")
    senha = input("senha: ")
    nome = checa_login(uid, senha)
    if nome == "":
        print("Login ou senha incorreto")
    else:
        print(saudacao(nome))
```

Exercício – sistema de *login*

```
import time
def saudacao(nome):
```

Exercício – sistema de *login*

```
import time
def saudacao(nome):
    hora = int(time.strftime('%H'))
```

Exercício – sistema de *login*

```
import time
def saudacao(nome):
    hora = int(time.strftime('%H'))
    if hora >= 6 and hora < 13:
        return "Bom dia, {}".format(nome)
```

Exercício – sistema de *login*

```
import time
def saudacao(nome):
    hora = int(time.strftime('%H'))
    if hora >= 6 and hora < 13:
        return "Bom dia, {}".format(nome)
    elif hora >= 13 and hora < 19:
        return "Boa tarde, {}".format(nome)
    else:
        return "Boa noite, {}".format(nome)
```

Exercício – sistema de *login*

```
import time
def main():
    nome = checa_login(input("username: "), input("senha: "))
    if nome == "":
        print("Login ou senha incorreto")
    else:
        print(saudacao(nome))

def saudacao(nome):
    hora = int(time.strftime('%H'))
    if hora >= 6 and hora < 13:
        return "Bom dia, {}".format(nome)
    elif hora >= 13 and hora < 19:
        return "Boa tarde, {}".format(nome)
    else:
        return "Boa noite, {}".format(nome)
```

Exercício – sistema de *login*

```
import time
def main():
    nome = checa_login(input("username: "), input("senha: "))
    print(saudacao(nome))

def saudacao(nome):
    if nome == "":
        return "Login ou senha incorreto"
    hora = int(time.strftime('%H'))
    if hora >= 6 and hora < 13:
        return "Bom dia, {}".format(nome)
    elif hora >= 13 and hora < 19:
        return "Boa tarde, {}".format(nome)
    else:
        return "Boa noite, {}".format(nome)
```

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

- $b \leq a$

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

- $b \leq a$
- $a > b$ or $a == b$

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

- $b \leq a$
- $a > b$ or $a == b$
- $a + 1 > b$

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

- $b \leq a$
- $a > b$ or $a == b$
- $a + 1 > b$
- $a > b - 1$

Exercício — condições

Dados os inteiros a e b , quais expressões são equivalentes a $a \geq b$?

- $b \leq a$
- $a > b$ or $a == b$
- $a + 1 > b$
- $a > b - 1$
- **not** $(a < b)$

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 25 a: 3

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 25 a: 3

x: 5

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3

x: 5

k > 0: True

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3

x: 5

k > 0: True

k: 20

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 saída: 35 x: 2

k > 0: True

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 saída: 35 x: 2

k > 0: True

k: 15 a: 38

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14**

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
    main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
    main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21**

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
    main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
    main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42**

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

k: 0

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

k: 0 a: 53

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

k: 0 a: 53 **saída: 28**

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
    x = a % 7
main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

k: 0 a: 53 **saída: 28** x: 4

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
    main()
```

k: 25 a: 3 x: 5

k > 0: True

k: 20 a: 23 **saída: 35** x: 2

k > 0: True

k: 15 a: 38 **saída: 14** x: 3

k > 0: True

k: 10 a: 48 **saída: 21** x: 6

k > 0: True

k: 5 a: 53 **saída: 42** x: 4

k > 0: True

k: 0 a: 53 **saída: 28** x: 4

k > 0: False

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: a: x:

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 25 a: x:

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x:

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 25 a: 3 x: 5

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 20 a: 3 x: 5

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 20 a: 23 x: 5

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 20 a: 23 x: 5

saída: 35

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 20 a: 23 x: 2

saída: 35

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 15 a: 23 x: 2

saída: 35

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 15 a: 38 x: 2

saída: 35

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 15 a: 38 x: 2

saída: 35

saída: 14

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 15 a: 38 x: 3

saída: 35

saída: 14

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 10 a: 38 x: 3

saída: 35

saída: 14

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 10 a: 48 x: 3

saída: 35

saída: 14

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 10 a: 48 x: 3

saída: 35

saída: 14

saída: 21

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 10 a: 48 x: 6

saída: 35

saída: 14

saída: 21

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 5 a: 48 x: 6

saída: 35

saída: 14

saída: 21

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 5 a: 53 x: 6

saída: 35

saída: 14

saída: 21

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():
    k = 25
    a = 3
    x = int(input("Digite um inteiro positivo: "))
    while (k > 0):
        k = k - 5
        a = a + k
        print(x * 7)
        x = a % 7
main()
```

k: 5 a: 53 x: 6

saída: 35

saída: 14

saída: 21

saída: 42

Exercício — pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 5 a: 53 x: 4

saída: 35

saída: 14

saída: 21

saída: 42

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 0 a: 53 x: 4

saída: 35

saída: 14

saída: 21

saída: 42

Exercício – pensando como o computador

O que este programa imprime se o usuário escolhe o número 5?

```
def main():  
    k = 25  
    a = 3  
    x = int(input("Digite um inteiro positivo: "))  
    while (k > 0):  
        k = k - 5  
        a = a + k  
        print(x * 7)  
        x = a % 7  
main()
```

k: 0 a: 53 x: 4

saída: 35

saída: 14

saída: 21

saída: 42

saída: 28

Exercício — soma de um trio Pitagórico

Três números inteiros positivos a , b e c , com $a < b < c$, formam um *trio Pitagórico* se $a^2 + b^2 = c^2$. Por exemplo, os números 3, 4 e 5 formam um trio Pitagórico pois $3^2 + 4^2 = 5^2$. Alguns números inteiros positivos podem ser escritos como a soma de um trio Pitagórico. Por exemplo, 12 é um desses números, pois $3 + 4 + 5 = 12$.

Escreva um programa que lê um número inteiro positivo e verifica se ele corresponde à soma de um trio pitagórico. O programa deve imprimir os números que compõem o trio ou uma mensagem informando que o número não é a soma de nenhum trio pitagórico.

Exercício – soma de um trio Pitagórico

Três números inteiros positivos a , b e c , com $a < b < c$, formam um *trio Pitagórico* se $a^2 + b^2 = c^2$. Por exemplo, os números 3, 4 e 5 formam um trio Pitagórico pois $3^2 + 4^2 = 5^2$. Alguns números inteiros positivos podem ser escritos como a soma de um trio Pitagórico. Por exemplo, 12 é um desses números, pois $3 + 4 + 5 = 12$.

Escreva um programa que lê um número inteiro positivo e verifica se ele corresponde à soma de um trio pitagórico. O programa deve imprimir os números que compõem o trio ou uma mensagem informando que o número não é a soma de nenhum trio pitagórico.

```
def pitagorico(a, b, c):
```

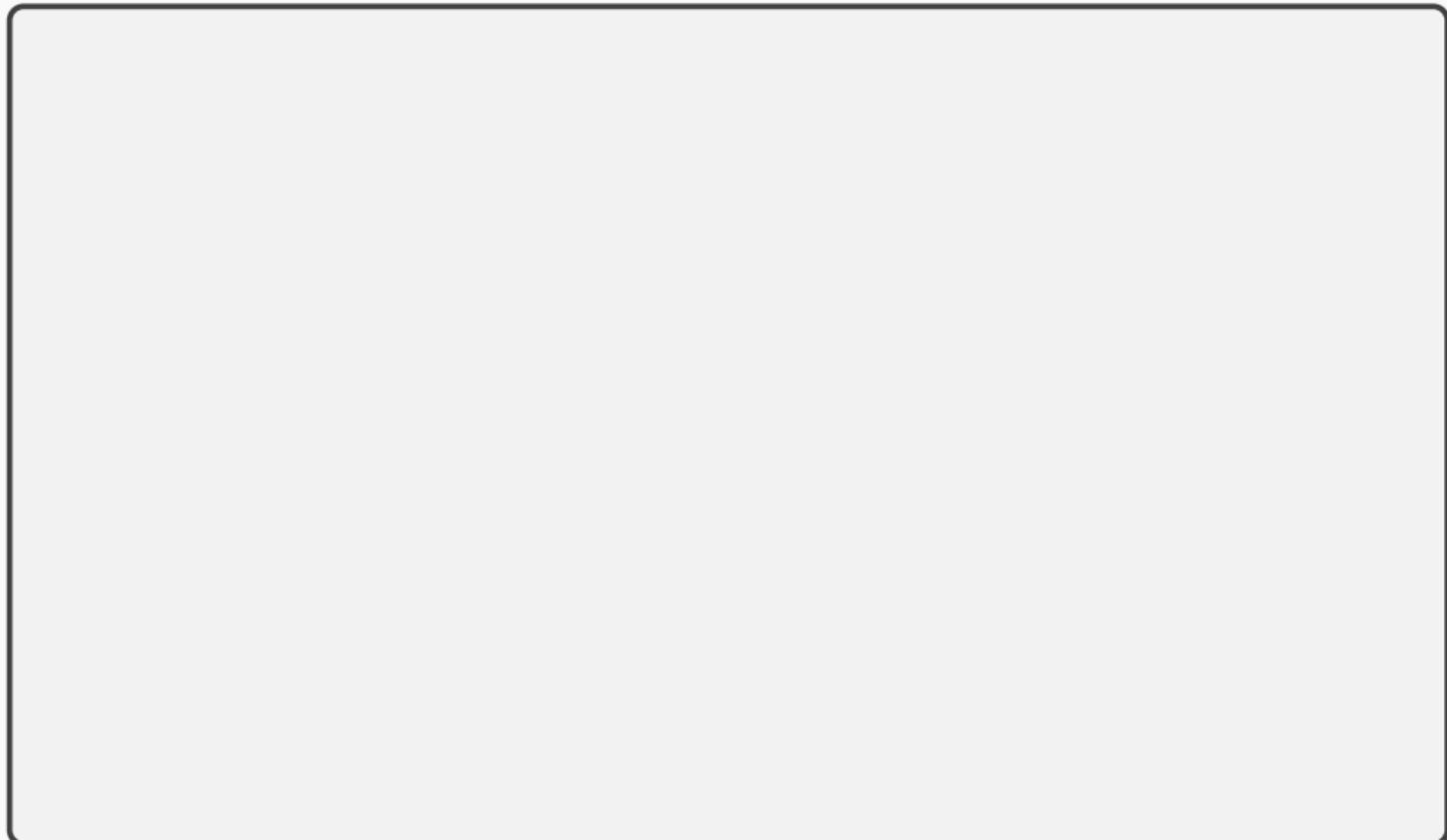
Exercício — soma de um trio Pitagórico

Três números inteiros positivos a , b e c , com $a < b < c$, formam um *trio Pitagórico* se $a^2 + b^2 = c^2$. Por exemplo, os números 3, 4 e 5 formam um trio Pitagórico pois $3^2 + 4^2 = 5^2$. Alguns números inteiros positivos podem ser escritos como a soma de um trio Pitagórico. Por exemplo, 12 é um desses números, pois $3 + 4 + 5 = 12$.

Escreva um programa que lê um número inteiro positivo e verifica se ele corresponde à soma de um trio pitagórico. O programa deve imprimir os números que compõem o trio ou uma mensagem informando que o número não é a soma de nenhum trio pitagórico.

```
def pitagorico(a, b, c):  
    return a**2 + b**2 == c**2
```

Exercício – soma de um trio Pitagórico



Exercício – soma de um trio Pitagórico

```
def main():
```

Exercício – soma de um trio Pitagórico

```
def main():  
    n = int(input("Digite um inteiro positivo: "))
```


Exercício – soma de um trio Pitagórico

```
def main():  
    n = int(input("Digite um inteiro positivo: "))  
  
    achei = False  
    while not achei  
  
        if achei:  
            print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))  
        else:  
            print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))

    achei = False
    while not achei and a < n:

        if achei:
            print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
        else:
            print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:

        if achei:
            print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
        else:
            print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:

            a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:

            b += 1

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:
            c = n - a - b

            b += 1

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:
            c = n - a - b
            achei = pitagorico(a, b, c)

            b += 1

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício — soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:
            c = n - a - b
            achei = pitagorico(a, b, c)
            if not achei:
                b += 1

        a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – soma de um trio Pitagórico

```
def main():
    n = int(input("Digite um inteiro positivo: "))
    a = 1
    achei = False
    while not achei and a < n:
        b = a + 1
        while not achei and a + b < n:
            c = n - a - b
            achei = pitagorico(a, b, c)
            if not achei:
                b += 1
        if not achei:
            a += 1

    if achei:
        print("O número {} é a soma do trio pitagórico {}, {} e {}".format(n, a, b, c))
    else:
        print("O número {} não é a soma de nenhum trio pitagórico".format(n))
```

Exercício – calculando π

Dado um inteiro $k \geq 0$, é possível calcular um valor aproximado de π através da expansão de Gregory-Leibniz:

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

Escreva um programa que lê um valor para k e realiza o cálculo acima.

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$



Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():  
    k = int(input("Digite o valor de k: "))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():  
    k = int(input("Digite o valor de k: "))  
  
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():  
    k = int(input("Digite o valor de k: "))  
  
    while n <= k:  
  
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():  
    k = int(input("Digite o valor de k: "))  
  
    n = 0  
    while n <= k:  
  
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():  
    k = int(input("Digite o valor de k: "))  
  
    n = 0  
    while n <= k:  
  
        n += 1  
        print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():
    k = int(input("Digite o valor de k: "))
    pi = 0

    n = 0
    while n <= k:

        n += 1
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():
    k = int(input("Digite o valor de k: "))
    pi = 0

    n = 0
    while n <= k:
        pi += 4 / (2*n + 1)

        n += 1
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():
    k = int(input("Digite o valor de k: "))
    pi = 0

    n = 0
    while n <= k:
        pi += sinal * 4 / (2*n + 1)

        n += 1
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – calculando π

$$\pi \approx \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots + \frac{(-1)^k 4}{2k+1}$$

```
def main():
    k = int(input("Digite o valor de k: "))
    pi = 0
    sinal = 1
    n = 0
    while n <= k:
        pi += sinal * 4 / (2*n + 1)
        sinal *= -1
        n += 1
    print("O valor de pi é aproximadamente {}".format(pi))
```

Exercício – Contando algarismos

Escreva uma função `contaAlgarismos(n, a)` que devolve o número de vezes que o algarismo `a` aparece no inteiro positivo `n` (zeros à esquerda obviamente não devem ser contados)

Exercício – Contando algarismos

Escreva uma função `contaAlgarismos(n, a)` que devolve o número de vezes que o algarismo `a` aparece no inteiro positivo `n` (zeros à esquerda obviamente não devem ser contados)

```
def contaAlgarismos(n, a):
```

Exercício – Contando algarismos

Escreva uma função `contaAlgarismos(n, a)` que devolve o número de vezes que o algarismo `a` aparece no inteiro positivo `n` (zeros à esquerda obviamente não devem ser contados)

```
def contaAlgarismos(n, a):
```

```
    contagem = 0
```

```
    while n > 0:
```

Exercício – Contando algarismos

Escreva uma função `contaAlgarismos(n, a)` que devolve o número de vezes que o algarismo `a` aparece no inteiro positivo `n` (zeros à esquerda obviamente não devem ser contados)

```
def contaAlgarismos(n, a):  
  
    contagem = 0  
    while n > 0:  
        dígito = n % 10  
        if dígito == a:  
            contagem += 1  
        n = n // 10  
    return contagem
```

Exercício – Contando algarismos

Escreva uma função `contaAlgarismos(n, a)` que devolve o número de vezes que o algarismo `a` aparece no inteiro positivo `n` (zeros à esquerda obviamente não devem ser contados)

```
def contaAlgarismos(n, a):  
    if n == 0 and a == 0:  
        return 1  
    contagem = 0  
    while n > 0:  
        dígito = n % 10  
        if dígito == a:  
            contagem += 1  
        n = n // 10  
    return contagem
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():  
    a = int(input("Digite o valor de a: "))  
    b = int(input("Digite o valor de b: "))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():  
    a = int(input("Digite o valor de a: "))  
    b = int(input("Digite o valor de b: "))  
  
    if permutação:  
        print("{} é uma permutação de {}".format(a, b))  
    else:  
        print("{} não é uma permutação de {}".format(a, b))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
    a = int(input("Digite o valor de a: "))
    b = int(input("Digite o valor de b: "))
    permutação = True

    if permutação:
        print("{} é uma permutação de {}".format(a, b))
    else:
        print("{} não é uma permutação de {}".format(a, b))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
    a = int(input("Digite o valor de a: "))
    b = int(input("Digite o valor de b: "))
    permutação = True
    n = 0
    while n <= 9

    if permutação:
        print("{} é uma permutação de {}".format(a, b))
    else:
        print("{} não é uma permutação de {}".format(a, b))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
    a = int(input("Digite o valor de a: "))
    b = int(input("Digite o valor de b: "))
    permutação = True
    n = 0
    while n <= 9

        n += 1
    if permutação:
        print("{} é uma permutação de {}".format(a, b))
    else:
        print("{} não é uma permutação de {}".format(a, b))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
    a = int(input("Digite o valor de a: "))
    b = int(input("Digite o valor de b: "))
    permutação = True
    n = 0
    while n <= 9:
        if contaAlgarismos(a, n) != contaAlgarismos(b, n):
            permutação = False
        n += 1
    if permutação:
        print("{} é uma permutação de {}".format(a, b))
    else:
        print("{} não é uma permutação de {}".format(a, b))
```

Exercício – Permutações

Um número a é uma *permutação* de um número b se é possível encontrar o número b reordenando os dígitos de a . Usando a função anterior, escreva um programa que lê dois números a e b e informa se a é uma permutação de b .

```
def main():
    a = int(input("Digite o valor de a: "))
    b = int(input("Digite o valor de b: "))
    permutação = True
    n = 0
    while n <= 9 and permutação:
        if contaAlgarismos(a, n) != contaAlgarismos(b, n):
            permutação = False
        n += 1
    if permutação:
        print("{} é uma permutação de {}".format(a, b))
    else:
        print("{} não é uma permutação de {}".format(a, b))
```