

Relatório: Trabalho 2

João Davi

Técnicas de Programação



Problema 1: Jogo de advinhação

- Programa que gere um número aleatório e o usuário deve acertar.
- Importa-se o método `randint` do módulo `random`.
- Input: Entrada do usuário com um número inteiro de 1 a 100.
- Uso do `if/elif`: Verificar o input.
- Output: Acerto ou erro, com dicas caso o usuário erre.

Implementação

Compara-se o número da entrada `numero_usuario` do usuário com o número do computador `numero_computador`.

Caso esteja certo, o programa encerrará, caso esteja errado será dada uma dica ao usuário e o programa continuará até o acerto.

O programa encerrará caso a entrada seja um valor negativo.

```
from random import randint

numero_computador = randint(1, 100)
tentativas = 0

while True:
    print(f"Tentativas : {tentativas}")
    numero_usuario = int(input("Digite um número inteiro entre 0 e 100: "))

    if numero_usuario < 0:
        print("Número negativo. Encerrando o programa.")
        break
    elif numero_usuario > 100:
        print("Digite um número válido! Entre 1 e 100.")
        continue
    elif numero_usuario == numero_computador:
        print(f"Parabéns, você acertou em {tentativas} tentativa(s)!")
        tentativas += 1
        break
    else:
        print("Você errou! Tente novamente. ", end='')
        tentativas += 1
        if numero_usuario > numero_computador:
            print(f"O número é menor que {numero_usuario}!")
        if numero_usuario < numero_computador:
            print(f"O número é maior que {numero_usuario}!")
        continue
```


Problema 2: Média dos números

- Programa que receba vários números até uma entrada negativa e calcule a média.
- Input Entrada do usuário com um número inteiro
- Uso do `if/elif/else`: Verificar o input
- Output: Imprime a média dos números.

Implementação

Compara se o número da entrada do usuário **numero** é maior que zero, para que o programa continue.

Caso seja maior que zero continua, se for igual a zero irá continuar no loop e pedirá outro número.

O programa encerrará caso a entrada seja um valor negativo.

```
media = 0.0
soma = 0
quantidade_numeros = 0

while True:
    numero = int(input("Digite um número: "))

    if numero < 0:
        print("Número negativo. Encerrando o programa.")
        break

    elif numero > 0:
        soma += numero
        quantidade_numeros += 1
        media = soma/quantidade_numeros
        print(f"A média é: {media:.2f}")

    else:
        print("Digite um número diferente de zero!")
```


Problema 3: Fatorial do número

- Programa que calcule o fatorial de um número.
- Input: Entrada do usuário com um número inteiro inteiro não negativo.
- Uso do `if`: Verificar se o número é não negativo.
- Uso do `for`: Calcular o fatorial do número.
- Output: Valor do fatorial.

Implementação

Compara se o número da entrada do usuário é menor que zero, para que o programa continue.

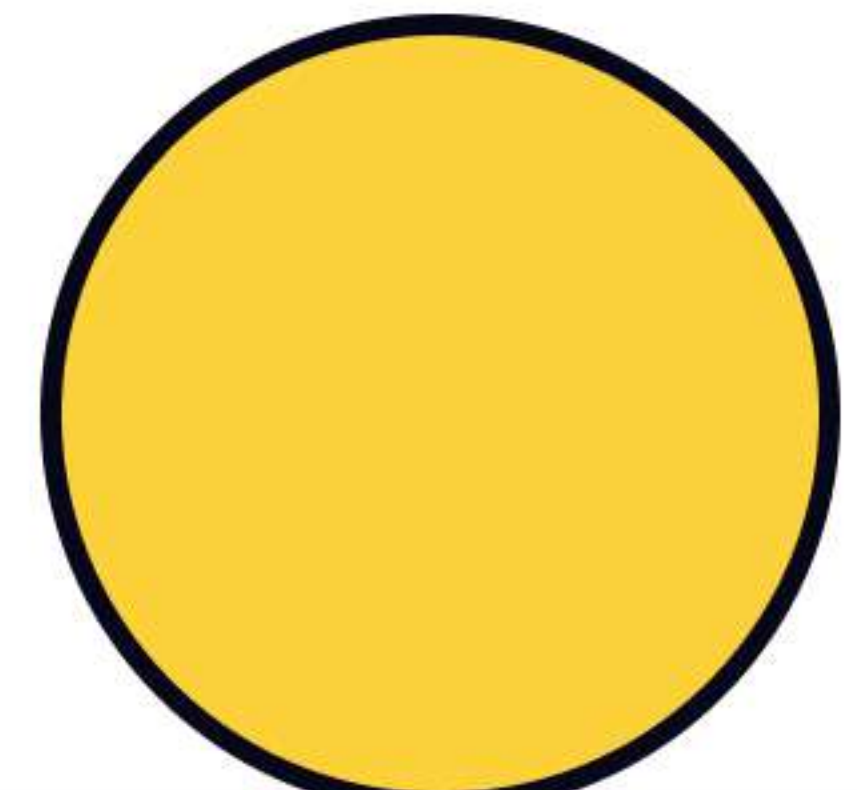
Caso seja maior que zero continua, percorrendo, através do **for**, pelo número e acumulando a multiplicação através da variável **fatorial**.

O programa repetirá até que um valor não negativo seja inserido.

```
while True:
    numero = int(input("Digite um número inteiro não negativo: "))
    if numero < 0:
        print("Digite um número válido!")
        continue
    break

fatorial = 1
print(f'{numero}! = ', end = '')

for i in range(numero, 0, -1):
    print(i, end = '')
    print(' x ' if i > 1 else ' = ', end = '')
    fatorial *= i
print(fatorial)
```



Problema 4: Número de vogais

- Programa que conte quantas vogais uma frase possui.
- Input: Entrada da frase do usuário.
- Uso do `lower()`: Tornar a string minúscula.
- Uso do `while`: Percorrer a string.
- Uso do `if`: Verificar se possui vogais em cada parte da frase dividida

Implementação

Percorre a string `frase` de acordo com seu tamanho e verifica a cada iteração se possui vogal na posição `i`.

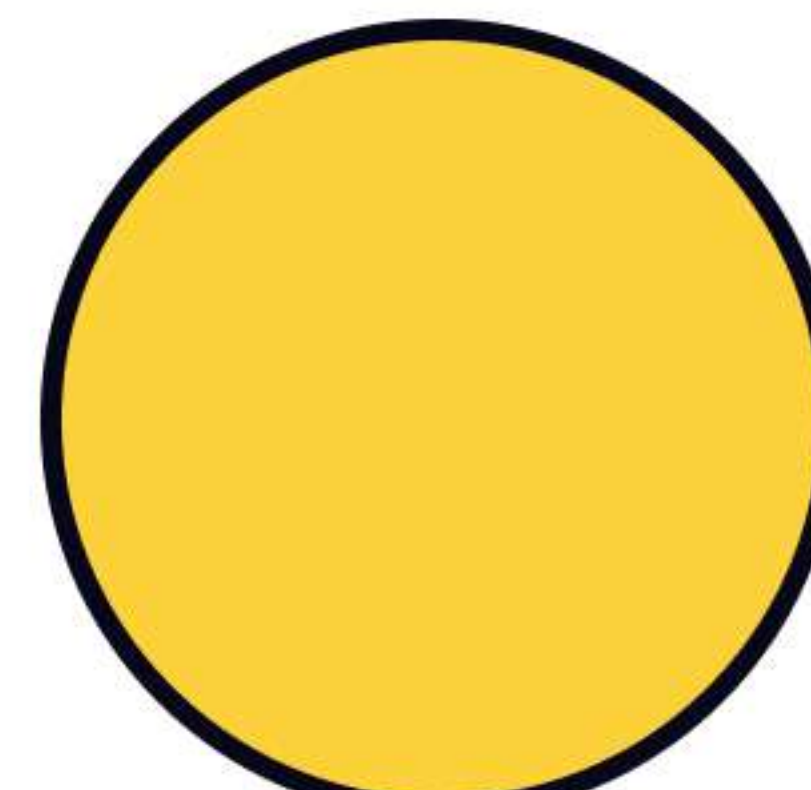
O contador `vogais` é incrementado caso a condição seja válida e o contador `i` também, para que saia do loop no final da string.

```
vogais = 0
i = 0

frase = str(input("Digite uma frase: ")).lower()

while i < len(frase):
    if frase[i] in "aeiou":
        vogais += 1
    i += 1

print(f"A frase tem {vogais} vogais")
```



Problema 5: Padrões em listas

- Programa que identifique padrões em uma lista de números fornecida pelo usuário.
- Input: Lista com números espaçados.
- Uso do `list comprehension`: Transformar a lista em uma lista de inteiros.
- Uso do `if/elif/else`: Verificar o padrão da lista.
- Output: Padrões presentes na lista.

Implementação

Converte-se a lista `lista` em uma lista de inteiros `lista_numeros` e usa-se o método `split()` para dividir a string, removendo os espaços.

Faz-se uma série de verificações para classificar os padrões presentes na lista. O `set()` transforma a lista em um conjunto, para analisarmos apenas os números únicos, já o `sorted()` usa-se para ordenar a lista.

```
lista = input("Digite uma lista de números separados por espaços: ")

lista_numeros = [int(numero) for numero in lista.split()]

if all(num % 2 != 0 for num in lista_numeros):
    print("Sequência de números ímpares")

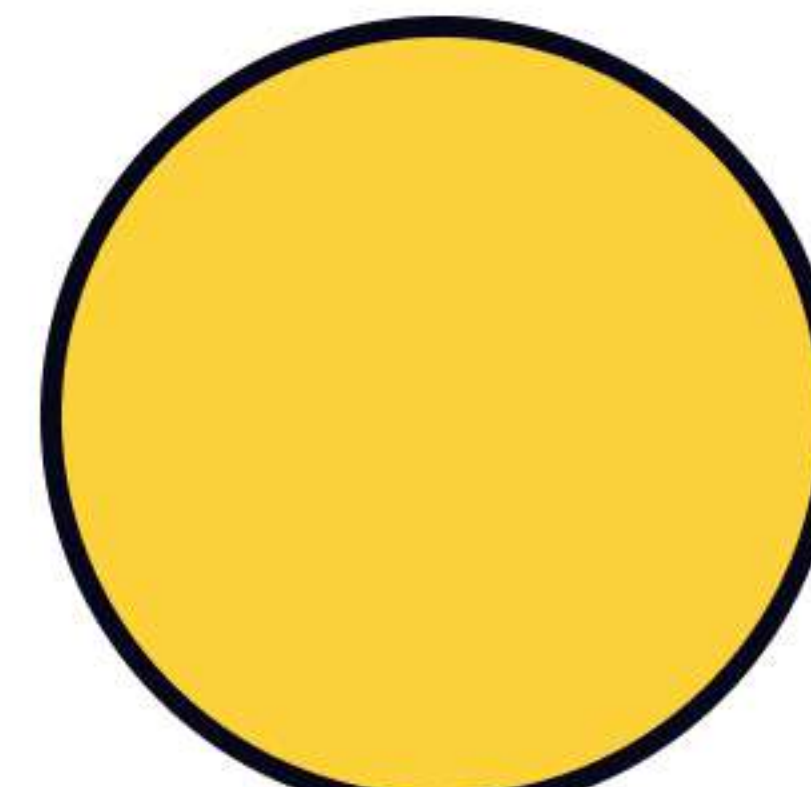
if all(num % 2 == 0 for num in lista_numeros):
    print("Sequência de números pares")

if len(set(lista_numeros)) != len(lista_numeros):
    print("Sequência de números repetidos")

elif lista_numeros == sorted(lista_numeros) and lista_numeros != lista_numeros[::-1]:
    print("Sequência de números crescente")

elif lista_numeros == sorted(lista_numeros, reverse=True) and lista_numeros != lista_numeros[::-1]:
    print("Sequência de números decrescente")

else:
    print("Não foi possível identificar um padrão na lista.")
```



Problema 6: Classificação de imagens

- Programa que percorre todas as imagens e conte quantas vezes cada animal apareceu.
- Criação de uma `lista`: Lista composta das aparições dos animais.
- Uso do `for`: Percorrer a lista criada.
- Output: Número de aparições de cada animal.

Implementação

Cria-se a lista `imagens_treinamento`, que possui uma o nome dos animais.

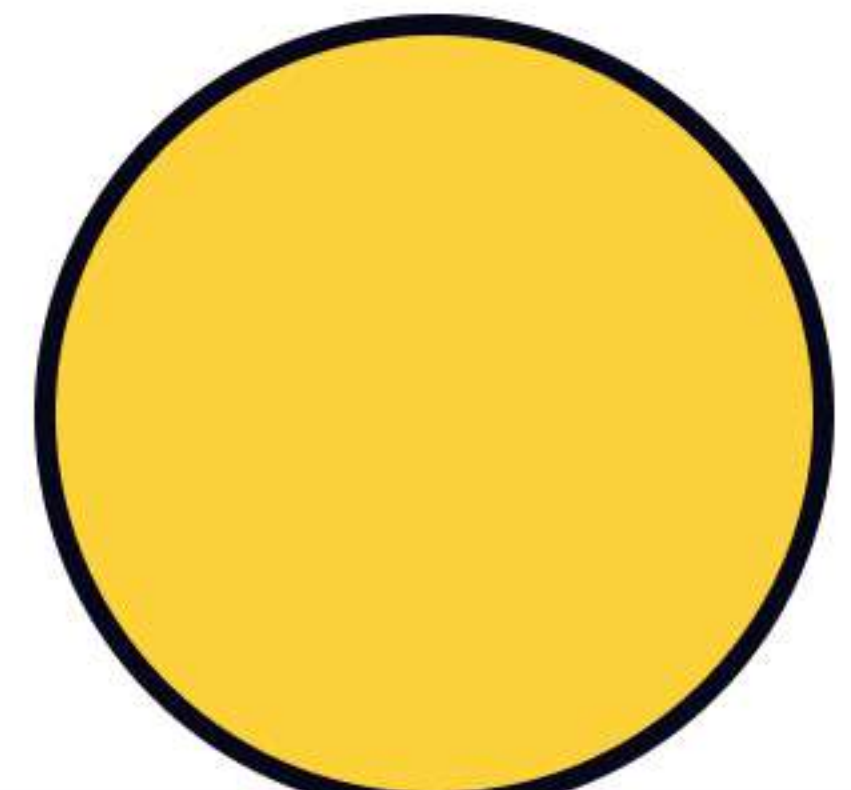
Percorre-se a lista Os contadores `gato`, `cachorro` e `passaro` incrementam caso o elemento na posição `i` seja o animal da condição.

```
imagens_treinamento = [
    "gato", "cachorro", "gato", "pássaro", "gato", "cachorro", "cachorro",
    "gato", "pássaro", "cachorro", "gato", "gato", "pássaro", "pássaro"
]

gato = cachorro = passaro = 0

for i in range(len(imagens_treinamento)):
    if imagens_treinamento[i] == "gato":
        gato += 1
    elif imagens_treinamento[i] == "cachorro":
        cachorro += 1
    else:
        passaro += 1

print(f"O gato apareceu {gato} vezes, o cachorro {cachorro} vezes e o pássaro {passaro} vezes.")
```



Problema 7: Tabuada de multiplicação

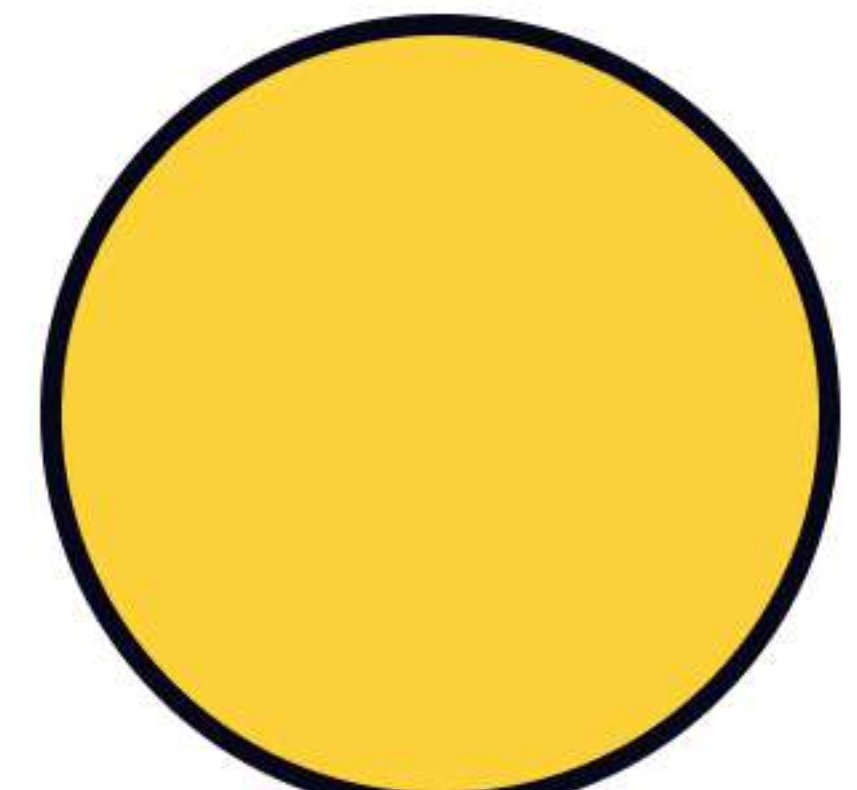
- Programa que exiba o número de linhas da tabuada de multiplicação que o usuário digitar.
- Input: Número da tabuada e número de linhas.
- Uso do `for`: Repetir a multiplicação conforme o número de linhas.
- Output: Tabuada de multiplicação

Implementação

Usa-se o laço `for` para iterar conforme o `numero_linhas` e imprime-se a tabuada de multiplicação do número `numero`.

```
numero = int(input("Digite um número para ver a tabuada: "))
numero_linhas = int(input("Digite o número de linhas da tabuada: "))

print(13*'---')
for i in range(1, numero_linhas + 1):
|   print(f"{numero} x {i} = {numero * i}")
print(13*'---')
```



Problema 8: Área de figuras

- Programa que calcule e exiba o valor da área das figuras do menu.
- Input: Opção da figura desejada.
- Uso do `if/elif/else`: Verificar qual opção foi escolhida e calcular a área de acordo com os valores da entrada do usuário.
- Output: Área da figura.

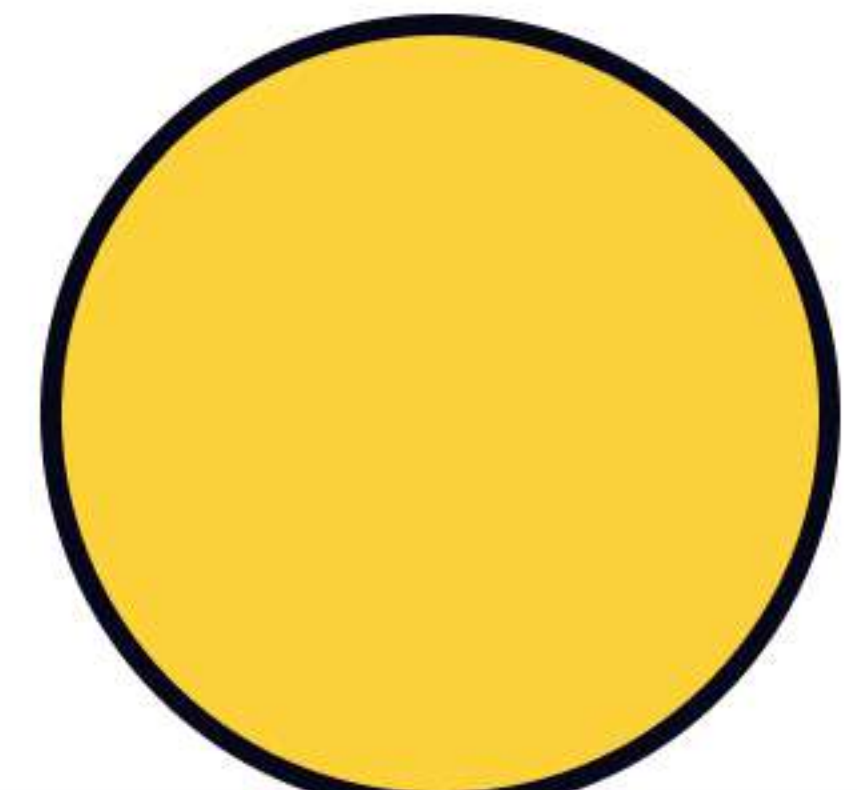
Implementação

Exibe-se um menu para o usuário escolher a figura, para calcular a área, que é calculada de formas diferente, de acordo com a **opção** escolhida.

Caso o usuário digite valores para a **opção** de 1 a 3, o programa irá continuar em loop, mas se o usuário digitar 4, o programa termina, saindo do loop.

```
while True:
    print(8 * '-=' + "MENU" + 8 * '-=')
    print("[1] - Quadrado\n[2] - Retângulo\n[3] - Triângulo\n[4] - Sair do programa")
    opcao = int(input("Escolha uma opção: "))

    if opcao == 1:
        lado = float(input("Digite o lado do quadrado: "))
        print(f"A área do quadrado é {lado ** 2:.2f}")
    elif opcao == 2:
        base = float(input("Digite a base do retângulo: "))
        altura = float(input("Digite a altura do retângulo: "))
        print(f"A área do retângulo é {base * altura}")
    elif opcao == 3:
        base = float(input("Digite a base do triângulo: "))
        altura = float(input("Digite a altura do triângulo: "))
        print(f"A área do triângulo é {(base * altura) / 2}")
    elif opcao == 4:
        print("FIM do programa.")
        break
    else:
        print("Opção inválida!")
```



Problema 9: Registro de usuário

- Programa que registra o email de um usuário, garantindo que seja único.
- Input: Email do usuário.
- Uso do `lower()/upper()`: Tornar a string minúscula/maiúscula.
- Uso do `strip()`: Remover os espaços vazios da string.
- Uso do `if`: Verificar se o email já foi cadastrado.
- Uso do `append()`: Adicionar o email cadastrado
- Output: Lista de emails cadastrados.

Implementação

Cria-se a lista `emails`, ao qual é adicionado uma string `email` caso não esteja ainda na lista.

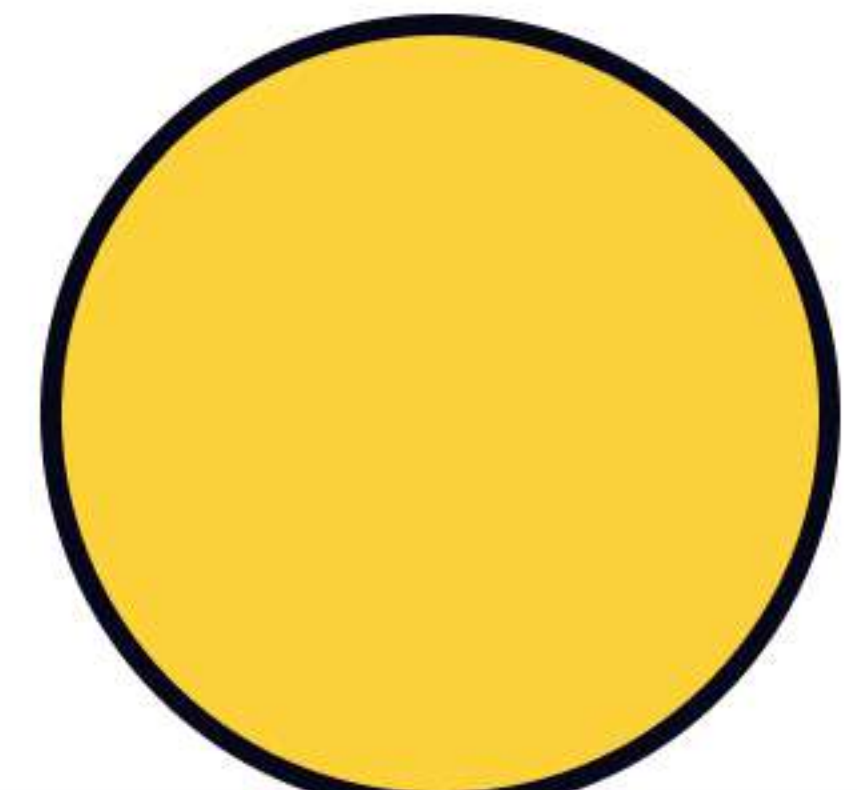
Usa-se os condicionais para adicionar o email à lista e para garantir que o programa continue, mas se o usuário não quiser continuar, o programa termina, saindo do loop e imprimindo os emails cadastrados.

```
emails = []
while True:
    email = input("Digite seu email: ").lower().strip()

    if email in emails:
        print("Email já cadastrado!")
    else:
        emails.append(email)
        print("Email cadastrado com sucesso!")

    continuar = input("Deseja continuar? [S/N] ").strip().upper()
    if continuar == "N":
        break

print(f"Emails cadastrados: {emails}")
```



Problema 10: Jogo da multiplicação

- Programa que gere um número aleatoriamente uma multiplicação e o usuário deverá acertar o resultado.
- Importa-se o método `randint` do módulo `random`.
- Input: Resposta da multiplicação.
- Uso do `if/elif/else`: Verificar se a resposta do usuário está certa.
- Output: Mensagem de acerto ou erro.

Implementação

Sorteia-se os valores para **numero1** e **numero2** para comparar com o resultado do usuário **resposta**.

Compara-se os valores, caso a resposta esteja certa, o programa continua com outra multiplicação, caso esteja errada continua com a mesma multiplicação até o usuário acertar.

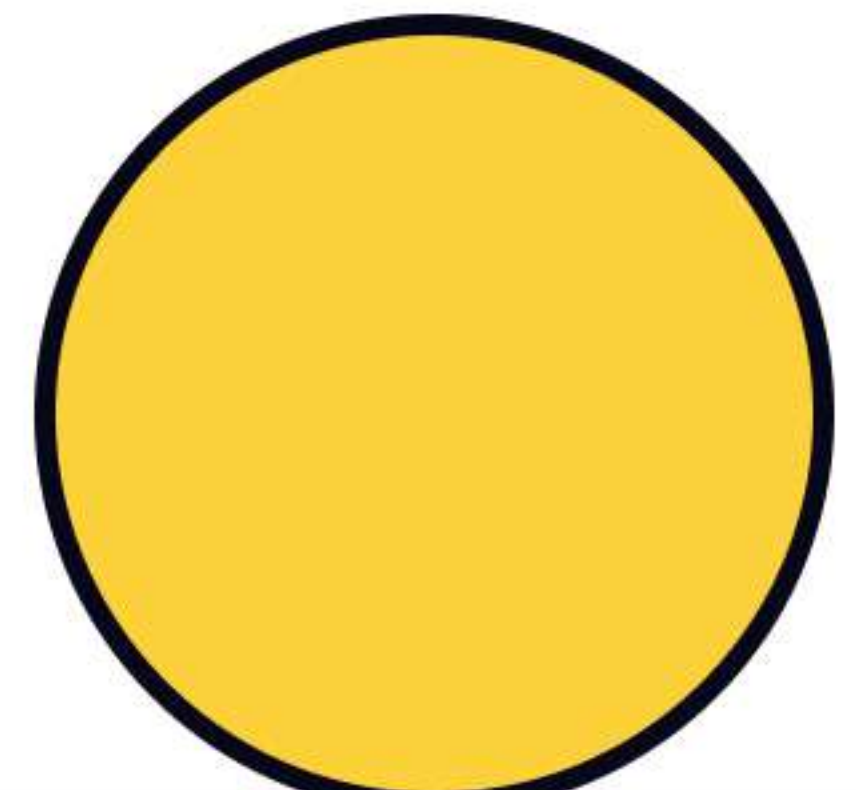
Fim do programa ao digitar 1.

```
from random import randint

while True:

    numero1 = randint(1, 10)
    numero2 = randint(1, 10)
    resposta = int(input(f"Quanto é {numero1} x {numero2}? (Digite 1 para sair) "))

    if resposta == 1:
        print("FIM do programa.")
        break
    elif resposta == numero1 * numero2:
        print("Resposta correta!")
    else:
        print("Resposta errada! Tente novamente.")
        while resposta != numero1 * numero2:
            resposta = int(input(f"Quanto é {numero1} x {numero2}? (Digite 1 para sair) "))
```



Email e Github



Email: `joaodaviliberato@alu.ufc.br`

Github:  `joaodaviliberato`

Repositório:  `Atividades_Python/Atividade2`