Serviço Nacional de Aprendizagem Industrial
SENAI
Desenvolvimento de Sistemas

E x e r c í c i o s R e s o l v i d o s

Codifique esses exercícios em Python, para cada exercício insira o significado do comando utilizado como comentário baseado no material das aulas anteriores, e se precisar, pesquise no Google. Posteriormente poste os códigos na atividade solicitada no AVA.

1- Elaborar um programa Python para calcular a soma de 1 até 50.

```
soma = 0
for i in range(1, 51):
    soma += i
    print("Soma:", soma)
```

2- Elaborar um programa Python para criptografar uma string utilizando a cifra de César.

```
deslocamento = int(input("Digite o deslocamento: "))
texto = input("Digite o texto a ser criptografado: ")
texto_criptografado = ""
for letra in texto:
    if letra.isupper():
        letra_criptografada = chr((ord(letra.lower()) + deslocamento - 97) % 26 + 65)
    elif letra.islower():
        letra_criptografada = chr((ord(letra) + deslocamento - 97) % 26 + 97)
    else: letra_criptografada = letra texto_criptografado += letra_criptografada
    print("Texto criptografado:", texto_criptografado)
```

3- Elaborar um programa Python para ler uma temperatura em Fahrenheint e converter para Celsius.

```
fahrenheit = float(input(“Digite a temperatura em Fahrenheit: “))
celsius = (fahrenheit - 32) * 5/9
print(“Temperatura em Celsius é:”, celsius)
```

4- Elaborar uma função para calcular o maior de três números.

```
def maior3(a, b, c):
    if a >= b and a >= c:
        return a
elif b >= c:
        return b
else: return c

n1=int(input(“Digite um número:”))
n2=int(input(“Digite um número:”))
n3=int(input(“Digite um número:”))

resultado = maior3(n1,n2,n3)
print(resultado)
```

5- Elaborar uma função recursiva em Python para calcular o MDC de dois números.

```
def mdc(a, b):
    if b == 0:
        return a
    else:
        return mdc(b, a % b)
```

```
num1 = int(input(“Digite um número:”))
num2 = int(input(“Digite outro número:”))

resultado = mdc(num1, num2)

print(“MDC:”, resultado
```

6- Elaborar uma função para converter de quilômetros para metros.

```
def converter_quilometros_para_metros(quilometros):
    metros = quilometros * 1000
    return metros
try:
quilometros = float(input(“Digite a distância em quilômetros: “))
metros = converter_quilometros_para_metros(quilometros)
print(“metros:”, metros)
except ValueError:
print(“Entrada inválida!”)
```

7- Elaborar uma função em Python para computar o maior e o menor elemento de uma lista.

```
def maior_menor(lista):
    maior = lista[0]
   menor = lista[0]
   for elemento in lista:
       if elemento > maior:
          maior = elemento
       elif elemento < menor:
           menor = elemento
```

```
    return maior, menor

lista=list()
i=1
while i<=10:
    elem = int(input("Digite um elemento da lista:"))
    lista.append(elem)
    i+=1
   print(lista)
maior, menor = maior_menor(lista)
print("Maior:", maior)
print("Menor:", menor)
```

8- Elaborar uma função para dobrar os elementos de uma lista.

```
def dobrar_lista(lista):
nova_lista = []
for elemento in lista:
    novo_elemento = elemento * 2
    nova_lista.append(novo_elemento)
return nova_lista

lista=list()
i=1
while i<=10:
   elem = int(input("Digite um elemento da lista:"))
   lista.append(elem)
    i+=1
print(lista)
nova_lista = dobrar_lista(lista)
```

```
print(nova_lista)
```

9- Elaborar um função em Python para ordenar uma lista utilizando a ordenação por inserção (Insertion Sort).

```
def insertion_sort(lista):
for i in range(1, len(lista)):
    chave = lista[i]
    j = i - 1
    while j >= 0 and chave < lista[j]:
        lista[j + 1] = lista[j]
        j -= 1 lista[j + 1] = chave

lista=list()
 i=1
while i<=10:
     elem = int(input("Digite um elemento da lista:"))
    lista.append(elem)
    i+=1
    print(lista)
    insertion_sort(lista)
    print(lista)
```

10- Elaborar um função para retornar o último elemento de uma lista.

```
def obter_ultimo_elemento(lista):
    if lista:
       return lista[-1]
    else: return None

lista=list()
```

```
i=1
while i<=5:
    elem = int(input("Digite um elemento da lista:"))
    lista.append(elem)
    i+=1
    print(lista)
    ultimo_elemento = obter_ultimo_elemento(lista)
    print("Último elemento da lista:", ultimo_elemento)
```

11- Elaborar em Python uma agenda de Contatos. Um contato tem os seguintes atributos: nome, telefone e e-mail.

```
class Contato:
def __init__(self, nome, endereco, email):
    self.nome = nome
    self.endereco = endereco
    self.email = email

class Agenda:
    def __init__(self):
        self.contatos = []
    def adicionar_contato(self, contato):
        self.contatos.append(contato)
    def remover_contato(self, contato):
        self.contatos.remove(contato)
    def listar_contatos(self):
        for contato in self.contatos:
            print("Nome:", contato.nome)
            print("Endereço:", contato.endereco)
            print("E-mail:", contato.email)

agenda = Agenda()
```

```
contato1 = Contato(“João”, “Rua A, 123”, “joao@example.com”)

contato2 = Contato(“Maria”, “Rua B, 456”, “maria@example.com”)
agenda.adicionar_contato(contato1)

agenda.adicionar_contato(contato2)

agenda.listar_contatos()

agenda.remover_contato(contato1)

agenda.listar_contatos()
```

12- Elaborar uma função para calcular o fatorial de um número utilizando recursão com cauda.

```
# A recursão de cauda é um tipo especial de recursão, no qual não existe processamento a ser feito depois de encerrada a chamada recursiva. Sendo assim, não é necessário guardar o estado do processamento no momento da chamada recursiva.

def fatorial(n, resultado=1):
 ‘’’
    Função que lê um número inteiro n >= 0 e imprime n!
‘’’
    if n == 0 or n == 1: # caso base
        return resultado
    else: # passo recursivo
        return fatorial(n - 1, n * resultado)
# Função principal
def main():
    n = int(input(“Digite um número inteiro: “))
    resultado = fatorial(n)
    print(20*”#”)
    print(“Fatorial:”, resultado)
   print(20*”#”) main()
```

13- Elaborar um programa Python para imprimir os números ímpares entre 1 e 100, inclusive.

```
for numero in range(1, 101):
    if numero % 2 != 0:
      print(numero,end=” “)
```

14- Elaborar um programa Python que leia uma lista com 10 inteiros e imprima a soma e média dos números

```
numeros = []
for i in range(10):
   try:
      numero = int(input(“Digite um número inteiro: “))
      numeros.append(numero)
     except ValueError:
    print(“Entrada inválida!!!”)
soma = sum(numeros)
media = soma / len(numeros)
print(“Soma:”, soma)
print(“Média:”, media)
```