'''-----------------------------------------------------------------------------

LISTA DE EXERCICIOS – INTROD. PROG. ESTRUTURADA – 2020/2

CURSO: Ciencia da Computação – CAMPUS: Cidade Univercitaria

PROFESSOR: Prof. Lauro Tomiatti para entrega é até o dia 05/11/2020

ALUNO: Luan Fernando Duarte Guimarães RA: N5804F-0 – SEMESTRE: 2º Semestre

ALUNO: Rafael da Silva Costa RA: N61543-5 – SEMESTRE: 2º Semestre

-----------------------------------------------------------------------------

'''

01 – Desenvolva uma classe que apresente todos os números primos existentes entre N1 e N2, em que N1

e N2 são números naturais fornecidos. Um número é primo quando é divisível somente por ele e pela unidade

(1).

numero = int(input('Digite um numero:'))

divisores = 0

cont = int(input('Até: '))

while cont >= numero:

for divisor in range(1, numero):

if numero % divisor == 0:

divisores = divisores + 1

if divisores > 1:

break

if divisores > 1:

print(numero,'não é primo')

elif numero == 1:

print(numero,'não é primo')

else:

print(numero,'é primo')

divisores = 0

numero = numero + 1

**02** – Escrever um algoritmo que produza na tela um triângulo de Pascal de grau “n” usando uma matriz. Abaixo temos um triângulo de Pascal de grau 6 (isto é, com seis linhas):

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

Os elementos extremos em cada linha são iguais a 1. Os outros são obtidos somando-se os dois valores que aparecem imediatamente acima e à esquerda na linha anterior. Exemplo: O quarto elemento da linha corresponde a soma do quarto e do terceiro elemento na linha anterior, isto é, 4 = 1 + 3.

def triangulo\_pascal(n):

for linha in range(1, n+1):

num = 1

linha\_atual = []

for i in range(1, linha+1):

linha\_atual.append(int(num))

num = num \* (linha-i) / i

print(linha\_atual)

triangulo\_pascal(10)

**03** – Escreva uma classe que leia um vetor de 30 posições de números inteiros e imprimir, logo após, gerar 2 vetores a partir dele, um contendo os elementos de posições ímpares do vetor e o outro os elementos pares. Imprimi-los no final.

lista = []

while len(lista) < 5:

num = int(input('Digite um numero: '))

if num in lista:

print('Nao e possivel adicionar numeros repetidos na lista, no caso "{num}"')

elif num not in lista:

lista.append(num)

print(lista)

lista\_pares = []

lista\_impares = []

for num in lista:

if num % 2 == 0:

lista\_pares.append(num)

elif num % 2 != 0:

lista\_impares.append(num)

print('Lista de numeros pares: {lista\_pares}')

print('Lista de numeros impares: {lista\_impares}')

**04** – Escreva um procedimento que receba um número arábico inteiro e imprima o corresponde número em romano. Por exemplo, para 5 a saída desejada é “V”. A função deve ser capaz de gerar o número romano para os 50 primeiros inteiros. Uma mensagem de erro deve ser mostrada caso um número fora dessa faixa seja recebido. Crie também um algoritmo que leia um valor inteiro e chame o procedimento criado acima para a impressão do número romano.

numero\_romano = []

while True:

num = int(input('Digite o numero a ser convertido: '))

if num > 50:

print(f'Nao e possivel conversao do numero "{num}"')

else:

break

dicionario\_romano = {1: 'I', 2: 'II', 3: 'III', 4: 'IV', 5: 'V', 6: 'VI', 7: 'VII', 8: 'VIII', 9: 'IX'}

while True:

if 10 <= num < 40:

numero\_romano.append('X' \* int(num/10))

num -= (int(num/10)) \* 10

if 40 <= num < 50:

numero\_romano.append('XL')

num -= 40

if num == 50:

numero\_romano.append('L')

num -= 50

if num < 10:

for item in dicionario\_romano.items():

if num == item[0]:

numero\_romano.append(item[1])

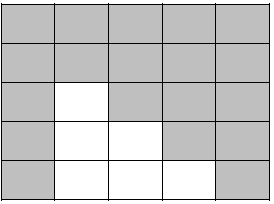
num = 0

if num == 0:

break

print(''.join(numero\_romano))

**05** – Preencher por leitura uma matriz M (5,5). Em seguida, calcular e imprimir a matriz toda e a média dos elementos das áreas assinaladas abaixo:

  
  
  
def matriz():

matriz = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]

for l in range(len(matriz)):

for c in range(len(matriz)):

matriz[l][c] = int(input(f'Entre com o valor para [{l},{c}] :'))

print('-=' \* 30)

for l in range(len(matriz)):

for c in range(len(matriz)):

print(f'[{matriz[l][c]}]', end= '')

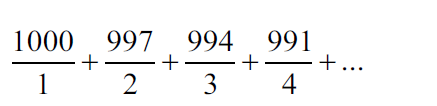
print()

media = (matriz[0][0] + matriz[0][1] + matriz[0][2] + matriz[0][3] + matriz[0][4] + matriz[1][0] + matriz[1][1] + matriz[1][2] + matriz[1][3] + matriz[1][4] + matriz[2][0] + matriz[2][2] + matriz[2][3] + matriz[2][4] + matriz[3][0] + matriz[3][3] + matriz[3][4] + matriz[4][0] + matriz[4][4]) / 19

print(media)

matriz()

06 – Fazer um algoritmo que calcule e escreva a soma dos 50 primeiros termos das seguintes séries:



num = 1000

div = 1

total = 0

for \_ in range(50):

res = num / div

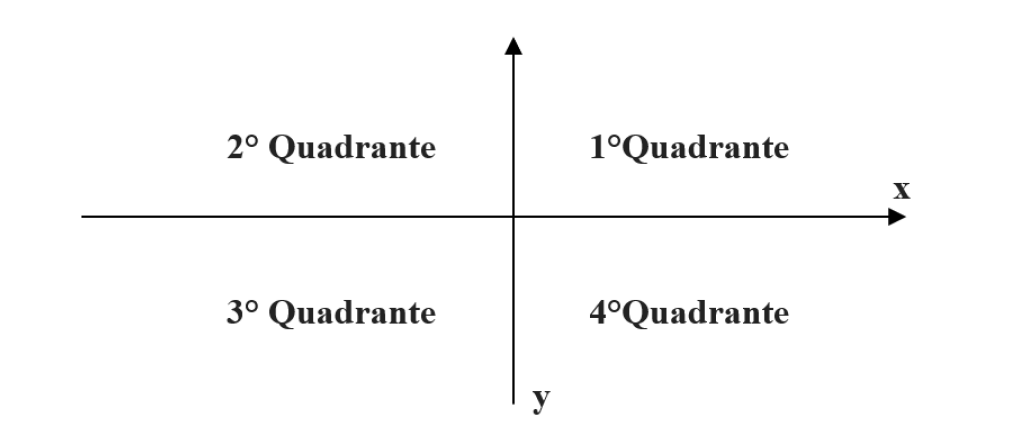
total += res

num = num - 3

div = div + 1

print(total)

**07** – Escrever um algoritmo que lê um par de coordenadas (x,y) inteiras e imprima uma mensagem informando em qual quadrante está o ponto. O algoritmo deve também ser capaz de identificar se o ponto está sobre um dos eixos ou no ponto central.

  
  
x = float(input('Digite o valor de "X": '))

y = float(input('Digite o valor de "Y": '))

print(f'X: {x}, Y: {y}')

if x != 0 and y != 0:

if x > 0 and y > 0:

print('Ponto ({x},{y}) esta no 1 quadrante')

if x > 0 and y < 0:

print('Ponto ({x},{y}) esta no 4 quadrante')

if x < 0 and y > 0:

print('Ponto ({x},{y}) esta no 2 quadrante')

if x < 0 and y < 0:

print('Ponto ({x},{y}) esta no 3 quadrante')

elif x == 0 and y != 0:

if y > 0:

print('Ponto ({x},{y}) esta sobre o eixo X')

print('Entre o 1 e 2 quadrante')

if y < 0:

print('Ponto ({x},{y}) esta sobre o eixo X')

print('Entre o 3 e 4 quadrante')

elif y == 0 and x != 0:

if x > 0:

print('Ponto ({x},{y}) esta sobre o eixo Y.')

print('Entre o 1 e 4 quadrante')

if x < 0:

print('Ponto esta sobre o eixo Y.')

print('Entre o 2 e 3 quadrante')

else:

print('Ponto ({x},{y}) esta no centro do plano cartesiano')

**08** – Crie um programa que leia um número entre 2 e 20 e gere uma tela com a seguinte configuração:

Digite um número: 7

Saída do programa:

1234567

x123456

xx12345

xxx1234

xxxx123

xxxxx12

xxxxxx1

while True:

num = int(input('Digite um numero entre 2 e 20: '))

if 2 < num < 20:

break

pass

numeros = ''

for numero in range(1, num+1):

numeros += str(numero)

print(numeros)

novo = ''

cont = len(numeros) - 1

for numero in range(1, len(numeros)):

novo = 'x' \* numero + numeros[0:cont]

cont -= 1

print(novo)

**09** – Faça uma função que recebe um valor inteiro e verifica se o valor é positivo, negativo ou zero. A função deve retornar 1 para valores positivos, ‐1 para negativos e 0 para o valor 0.

def calc():

num = int(input('Entre com o numero: '))

if num < 0 :

print('valor negativo: ',-1)

elif num > 0:

print('valor positivo:',1)

else:

print('O valor é zero:',0)

return calc()

calc()

**10** – Elabore um algoritmo que leia dois vetores de dez posições e faça a multiplicação dos elementos da seguinte forma: o primeiro do vetor 1 com o último do vetor 2, o segundo do vetor 1 com o penúltimo do vetor 2 e assim por diante, colocando o resultado num terceiro vetor, que deve ser mostrado como saída.

vet1 = []

vet2 = []

for num in range(10):

vet1.append(input('Digite o {num+1} numero do 1 vetor: '))

for num in range(10):

vet2.append(input('Digite o {num+1} numero do 2 vetor: '))

print(vet1)

print(vet2)

vet\_soma = []

contador1 = 0

contador2 = len(vet2)-1

for numero in vet1:

vet\_soma.append(int(vet1[contador1]) + int(vet2[contador2]))

contador1 += 1

contador2 -= 1

print('Lista apos soma: {vet\_soma}')

**11** – Construa um programa que, para um grupo de 50 valores inteiros, determine:

a) A soma dos números positivos;

b) A quantidade de valores negativos;

def soma\_lista(som):

total = 0

for i in som:

total= total+i

return total

som=[]

neg=[]

resultado=0

contador=0

while contador <=50:

num=float(input('Digite um total de 50 numeros: \n'))

if num >=0:

som.append(num)

print ('\nOs numeros positivos são:\n',som )

if num <0:

neg.append(num)

print('\nOs numeros negativos são:\n', neg)

print('\nA soma dos números positivos é:\n ',soma\_lista(som))

contador+1

**12** - Desenvolva um programa que receba 25 números (tipo float) digitados pelo usuário e apresente no final a quantidade de números positivos, negativos, zeros, pares e ímpares digitados.

pos=[]

neg=[]

zeros=[]

pares=[]

impares=[]

count= 0

while count <= 25:

num =float (input('Digite um numero real qualquer: '))

if num > 0:

pos.append(num)

elif num < 0:

neg.append(num)

elif num == 0 :

zeros.append(num)

if (num%2) == 0:

pares.append(num)

if num%2:

impares.append(num)

print('\n Numeros Positivos: \n',pos)

print('\n Numeros Negativos: \n',neg)

print('\n Quatidade de zeros: \n',zeros)

print('\n Numeros pares: \n',pares)

print('\n Numeros impares: \n',impares)

count +=1

**13** - Desenvolva um programa que receba nome, idade e salário digitados pelo usuário e apresente no final quantas dessas idades estão entre 15 e 17 anos, quantas são maiores de 21 anos, quantos salários estão entre R$1.500,00 e R$2.000,00, quantos estão acima de R$3.500,00 e qual é o maior e o menor salário digitado. (utilizar laço de repetição com opção de saída do sistema).

def cargos():

if idade >= 15 and idade <=17:

menor.append(nome)

return

def maiores():

if idade >= 21:

maior.append(nome)

return

def sala():

if salario >= 1500 and salario <= 2000:

sal.append(salario)

return

def dinheirao():

if salario >= 3500:

money.append(salario)

return

menor=[]

maior=[]

sal=[]

money=[]

op='N'

while op!='S' and op!='s':

nome = input('Insira um nome: ')

idade = int(input('Insira a idade: '))

salario = float(input('Insira o salario: '))

cargos()

maiores()

sala()

dinheirao()

print('\nOs menores de idade (15 ~ 17) são: ''\n', menor)

print('\nOs maiores de idade (21+) são: ''\n', maior)

print('\n Os com baixa renda (1500 ~ 2000) tem: ''\n',sal)

print('\n Os com melhor renda (3500+) tem um salario de: ''\n',money)

op=input('\nPara sair do programa digite' 'S: ')

**14** – Desenvolva um programa que receba nome e salário de um funcionário e calcule o valor do salário líquido desse funcionário, utilizando função, descontando os impostos INSS e Imposto de Renda (IR) conforme tabela oficial vigente. (utilizar laço de repetição com opção de saída do sistema).

lista\_empregados = []

lista = []

while True:

lista.append(input('Digite o nome: '))

lista.append(input('Digite o salario: '))

lista\_empregados.append(lista)

lista = []

teste = input('Digite "sair" caso deseje parar: ')

if teste == 'sair':

break

else:

pass

print(f'\n{lista\_empregados}\n')

for nome, salario in lista\_empregados:

salarioNovo = salario

print('Salario bruto de {nome}: {float(salario)}')

if float(salario) <= 1045.00:

salarioNovo = float(salario) - float(salario) \* 0.075

print('Salario de {nome} apos INSS: {salarioNovo}')

if 1045.01 <= float(salario) <= 2089.60:

salarioNovo = float(salario) - float(salario) \* 0.09

print('Salario de {nome} apos INSS: {salarioNovo}')

if 2089.61 <= float(salario) <= 3134.40:

salarioNovo = float(salario) - float(salario) \* 0.12

print('Salario de {nome} apos INSS: {salarioNovo}')

if float(salario) > 3134.40:

salarioNovo = float(salario) - float(salario) \* 0.14

print('Salario de {nome} apos INSS: {salarioNovo}')

if float(salario) <= 1903.98:

salarioNovo = float(salarioNovo) - float(salario) \* 0

print('Salario de {nome} apos INSS e IR: {salarioNovo}')

if 1903.99 <= float(salario) <= 2826.65:

salarioNovo = float(salarioNovo) - float(salario) \* 0.075

print('Salario de {nome} apos INSS e IR: {salarioNovo}')

if 2826.66 <= float(salario) <= 3751.05:

salarioNovo = float(salarioNovo) - float(salario) \* 0.15

print('Salario de {nome} apos INSS e IR: {salarioNovo}')

if 3751.06 <= float(salario) <= 4664.68:

salarioNovo = float(salarioNovo) - float(salario) \* 0.225

print('Salario de {nome} apos INSS e IR: {salarioNovo}')

if float(salario) > 4664.68:

salarioNovo = float(salarioNovo) - float(salario) \* 0.275

print('Salario de {nome} apos INSS e IR: {salarioNovo}')

print()

**15** – Elabore um programa que receba dois números (tipo float) digitados pelo usuário e pergunte qual operação ele deseja realizar. Operações possíveis: soma, subtração, multiplicação, divisão, maior e menor número. Exiba no final os números digitados e o resultado da operação escolhida.

r= 'N'

while r != 'S' or r != 's':

n1=float(input('Digite o Primeiro número: '))

n2=float(input('Digite o Segundo número: '))

op=input('Digite a operação desejada: ')

if op == '1':

print(n1+n2)

elif op == '2':

print(n1-n2)

elif op == '3':

print(n1\*n2)

elif op == '4':

print(n1/n2)

elif op == '5':

n1 < n2

print(n2)

elif op == '6':

n1 > n2

print(n1)

r=input('Se desejar sair do código digite "S":')

**16** – Elabore um programa que faça a conversão de moedas. O programa deve receber uma quantidade em determinada moeda e a taxa de conversão e apresentar a quantidade convertida na moeda selecionada. Conversões possíveis: dólar para real, euro para real, real para dólar e real para euro. (utilizar laço de repetição com opção de saída do sistema).

op = 'N'

while op != 'S' or op != 's':

mon=float(input('Digite o valor em dinheiro que gostaria de converter:\n '))

con=input('Selecione o método de conversão desejado:\n 1= Real para dolar;\n 2= Real para euro;\n 3= Dolar para Real;\n 4= Dolar para euro;\n 5= Euro para Real;\n 6= Euro para Dolar;\n')

if con == '1':

mon = mon\*5.74

print('A Conversão de Real para Dolar ficou:\n',mon)

if con == '2':

mon = mon\*6.71

print('A conversão de Real para Euro ficou:\n ',mon)

if con == '3':

mon= mon\*0.17

print('A conversão de Dolar para Real ficou:\n',mon )

if con == '4':

mon= mon\*0.86

print('A conversão de Dolar para Euro ficou:\n ',mon)

if con == '5':

mon= mon\*0.15

print('A Conversão de Euro para Real ficou:\n',mon)

if con == '6':

mon= mon\*1.17

print('A Conversão de Euro para Dolar ficou:\n',mon)

op=input('Para sair do programa digite "S", Para reutilizar o programa, digite "n":\n')

**17** – Desenvolver um programa que calcule a média aritmética simples das notas de um aluno com opção de escolha para entrada de 2 notas, 3 notas ou 4 notas. Exiba no final o nome do aluno e sua média com a informação: “Aprovado” se média maior ou igual a 7, “Reprovado” se média menor que 4 e “Exame” nos demais casos. (utilizar laço de repetição com opção de saída do sistema).

op='n'

while op !='S' or op!='s':

notas=input('Deseja utilizar quantas notas para a média?\n 1 - Para 2 notas;\n 2- Para 3 notas;\n 3- Para 4 Notas \n: ')

if notas == '1':

n1=int(input('\nInsira a nota da N1:\n '))

n2=int(input('\nInsira a nota da N2:\n '))

media= n1 + n2

resultado = media / 2

print('Sua média é: ',resultado)

elif notas == '2':

n1=int(input('\nInsira a nota da N1:\n '))

n2=int(input('\nInsira a nota da N2:\n '))

n3=int(input('\nInsira a nota da N3:\n '))

media= n1+n2+n3

resultado= media/3

print('Sua média é: ',resultado)

elif notas == '3':

n1=int(input('\nInsira a nota da N1:\n '))

n2=int(input('\nInsira a nota da N2:\n '))

n3=int(input('\nInsira a nota da N3:\n '))

n4=int(input('\nInsira a nota da N4:\n '))

media= n1+n2+n3+n4

resultado= media/4

print('Sua média é: ',resultado)

if resultado >=7:

print('\nVocê foi aprovado com média: ',resultado)

elif resultado < 4:

print('\nVocê foi reprovado, sua média foi de: ',resultado)

else:

print('\nVocê está de exame, sua nota é de: ',resultado)

op=input('\nSe desejar sair do programa, aperte "S", aperte qualquer outra tecla para continuar:\n ')

**18** – Desenvolver um programa que entre com as notas (NP1 e NP2), quantidade de falta e carga horaria da disciplina e informe se o aluno “Passou Direto”, “Exame”, “Substitutiva” ou “Reprovado”. Caso o aluno entre com “NC” o aluno deve realizar a PSUB. Caso o aluno fique com nota insatisfatória, deve realizar um exame e após o lançamento, o programa deve reanalisar a situação, acrescentando “Aprovado após exame” ou “Reprovado após exame”. A regra deve ser a mesma do Manual do Aluno (vide páginas 14 e 25) disponível no link: https://unip.br/presencial/servicos/aluno/download/calendario\_manual\_cursos\_tradicionais1.pdf

carga = int(input('Insira a carga horaria semestral da materia\n'))

maximo = carga / 4

print(f'Voce poderia ter faltado ate {int(maximo)} vezes nesta materia')

faltas = int(input('Numero de faltas: '))

if faltas <= maximo:

while True:

while True:

np1 = float(input('NP1: '))

if np1 < 0 or np1 > 10:

print('As notas devem possuir um valor positivo entre 0 e 10...')

pass

else:

break

while True:

np2 = float(input('NP2: '))

if np2 < 0 or np2 > 10:

print('As notas devem possuir um valor positivo entre 0 e 10...')

pass

else:

break

break

if np1 == 0 and np2 == 0:

teste = 0

while True:

print('Apenas uma das provas pode ser substituida\n'

'1 - NP1\n'

'2 - NP2')

escolha = int(input('Qual prova deseja substituir?\n'))

if escolha == 1:

pergunta = str(input('O aluno realizou a NP1?\n'

'Responda com "sim" ou "nao": '))

if pergunta == 'sim':

print('O aluno nao pode substituir esta prova...')

np1 = 0

teste += 1

if pergunta == 'nao':

while True:

sub\_np1 = float(input('Prova sub NP1: '))

if sub\_np1 < 0 or sub\_np1 > 10:

print(f'{sub\_np1} nao e uma nota valida...')

pass

else:

break

np1 = sub\_np1

break

if escolha == 2:

pergunta = str(input('O aluno realizou a NP2?\n'

'Responda com "sim" ou "nao": '))

if pergunta == 'sim':

print('O aluno nao pode substituir esta prova...')

np2 = 0

teste += 1

if pergunta == 'nao':

while True:

sub\_np2 = float(input('Prova sub NP2: '))

if sub\_np2 < 0 or sub\_np2 > 10:

print(f'{sub\_np2} nao e uma nota valida...')

pass

else:

break

np2 = sub\_np2

break

if teste == 2:

print('Entao, nenhuma das notas pode ser substituida')

break

if escolha not in [1, 2]:

print('Opcao nao existe...')

pass

elif np1 == 0 and np2 != 0:

pergunta = str(input('O aluno realizou a NP1?\n'

'Responda com "sim" ou "nao": '))

if pergunta == 'sim':

np1 = 0

if pergunta == 'nao':

while True:

sub\_np1 = float(input('Prova sub NP1: '))

if sub\_np1 < 0 or sub\_np1 > 10:

print(f'{sub\_np1} nao e uma nota valida...')

pass

else:

break

np1 = sub\_np1

elif np2 == 0 and np1 != 0:

pergunta = str(input('O aluno realizou a NP2?\n'

'Responda com "sim" ou "nao": '))

if pergunta == 'sim':

np2 = 0

if pergunta == 'nao':

while True:

sub\_np2 = float(input('Prova sub NP2: '))

if sub\_np2 < 0 or sub\_np2 > 10:

print(f'{sub\_np2} nao e uma nota valida...')

pass

else:

break

np2 = sub\_np2

media = (np1 + np2)/2

print(f'NP1: {np1}\n'

f'NP2: {np2}\n'

f'Media: {media}')

if media >= 7:

print('Aluno aprovado!')

else:

print('Aluno de exame...')

while True:

exame = float(input('Insira a nota do exame: '))

if exame > 10 or exame < 0:

print('Nota invalida...')

pass

else:

break

if exame >= 10 - media:

print('Aluno aprovado apos exame')

else:

print(f'Aluno reprovado apos exame por {10 - (exame + media)} ponto(s)')

else:

print('Aluno reprovado por carga horaria insuficiente.')

**19** – Uma pista de Kart permite 10 voltas para cada um de 6 corredores. Escreva um programa que leia todos os tempos em segundos e os guarde em um dicionário, onde a chave é o nome do corredor. Ao final diga de quem foi a melhor volta da prova e em que volta; e ainda a classificação final em ordem (1o o campeão). O campeão é o que tem a menor média de tempos.   
  
corredores = []

for num in range(1, 7):

corredor = str(input(f'Digite o nome do {num} corredor(a): '))

corredores.append(corredor)

analise = {}

for corredor in corredores:

voltas = []

for volta in range(1, 11):

while True:

tempo = float(input(f'Insira o tempo em segundos da {volta} volta feito por {corredor}: '))

if tempo > 0:

voltas.append(tempo)

break

else:

print(f'{tempo} segundos inexistente...')

pass

analise.update({corredor: voltas})

print(f'\nDicionario com nome dos corredores e seus respectivos tempos:'

f'\n{analise}\n')

menor\_volta = analise.get(corredores[0])[0]

volta\_atual = 1

corredor\_recordista = corredores[0]

for corredor, voltas in analise.items():

contador = 1

for volta in voltas:

if float(volta) < float(menor\_volta):

corredor\_recordista = corredor

menor\_volta = float(volta)

volta\_atual = contador

contador += 1

print(f'Menor volta realizada por {corredor\_recordista}\n'

f'Realizada em {menor\_volta} segundos durante sua {volta\_atual} volta\n')

medias = []

for corredor, voltas in analise.items():

soma = 0

for volta in voltas:

soma += float(volta)

media = soma / len(voltas)

medias.append([media, corredor])

ordenada = sorted(medias)

contador = 1

for media, corredor in ordenada:

print(f'Em {contador} lugar: {corredor} com uma media de {media} segundos.')

contador += 1

**20** – Desenvolva um programa que receba 10 valores de temperatura ambiente de uma cidade. O programa deve armazenar tais valores em uma lista denominada temp, conforme os valores vão sendo recebidos. Após isso, o programa deve retornar a média de temperatura, temperatura máxima, temperatura mínima e por fim criar uma segunda lista (chamada dados) e por os valores em ordem crescente.

temp = [0,0,0,0,0,0,0,0,0,0]

print('---Entre com 10 temperaturas ambientes da cidade ---\n')

for l in range(len(temp)):

temp[l] = int(input(f'Entre com a temperatura ambiente da cidade {l+1}: '))

print('\n Lista de temperaturas:',temp)

temp.sort()

print('\n Lista de temperaturas em ordem crescente:',temp)

media = sum(temp) / len(temp)

print('\na média de temperatura é: ',media)

minima = min(temp)

print('\na minima de temperatura é: ',minima)

maxima = max(temp)

print('\na maxima de temperatura é: ',maxima)

dados = [maxima, minima ,media]

dados.sort()

print('\nDados ordenados em ordem crescente: ',dados)