

Universidade Federal do Pará  
Disciplina: Programação 1  
Tópicos: arquivos.  
Data: 06/03/25 à 13/03/25

**Não será permitido:**

- uso de break, while True, sum(), sort(), min(), max().

**Observação 1:**

Exemplo da criação de um dicionário com as chaves "Nome", "Matricula" e "Media":

```
aluno = {"Nome": "Joao das Neves", "Matricula": 88888888, "Media": 30.5}
```

Exemplo de acesso ao valor de uma chave do dicionário:

```
variavel = aluno["Nome"]
```

**Observação 2:**

Sintaxe para definição de subprogramas (procedimento ou função):

```
def nome_subprograma(lista_parametros):  
    #bloco de instruções  
    #caso o subprograma seja do tipo função utilizar o comando return
```

**Observação 3:**

Para ler os dados de um arquivo texto:

```
with open (arquivoDados) as arq:  
    for i in arq: #i é uma linha com \n
```

**Atividades**

1) Quais são os modos de abertura de um arquivo?

2) Como funciona o gerenciamento de arquivos em Python utilizando a instrução `with open()` e qual é a sua vantagem em relação ao uso do `open()` e `close()` separadamente?

3) Qual a diferença entre os métodos `read()`, `readline()` e `readlines()` ao ler arquivos de texto em Python, e em que situações cada um é mais adequado de ser utilizado?

4) Você tem uma lista de compras no formato de pares **item quantidade**. Sua tarefa é **completar** o código abaixo para ler esses dados de um arquivo de texto, onde cada linha conterá um item e a respectiva quantidade. Os dados lidos são armazenados em uma lista, onde cada elemento da lista é um dicionário.

Exemplo de conteúdo do arquivo de "entrada.txt":

```
ovo 10  
leite 3  
açucar 2  
arroz 2  
feijão 3  
café 1  
sabonete 4
```

Código a ser completado para ler os dados do arquivo "entrada.txt":

```
leitura="entrada.txt"
lista =[]

def lendo (leitura, lista):
    with open (leitura) as arq:
        #complete aqui
        for i in arq:
            lista.append({})
            #complete aqui

lendo (leitura,lista)
print(lista)
```

5) A partir do programa acima, acrescente instruções para imprimir na tela a quantidade total de itens a serem comprados, isto é, a soma das quantidades de cada produto.

6) Dados dois vetores X e Y, ambos com n elementos, determinar o produto escalar desses vetores. Para calcular o produto escalar utilize a seguinte fórmula:

$$A * B = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

Entrada:

Os dados dos vetores estão em um arquivo-texto. A primeira linha da entrada será o número dos elementos dos vetores. As linhas seguintes serão compostas pelos valores de cada posição primeiro vetor e em seguida, serão os valores do vetor Y.

Exemplo de conteúdo do arquivo de entrada:

```
5
2.2
3.5
1
2.8
1.6
1.2
6.5
32
1.65
```

Saída:

A saída será composta de apenas uma linha contendo o valor do produto escalar.

Exemplo de saída:

```
36.63
```

7) A moeda corrente de um determinado país possui notas de 100, 50, 20, 10, 5, 2 e 1 unidade monetária. Faça um programa que recebe um valor inteiro positivo nesta moeda, e retorna quantas notas de cada valor devem ser utilizadas para obter tal valor, utilizando o menor número de notas possível. As saídas devem ser armazenadas em um arquivo-texto chamado "notas.txt".

Entradas:

1. Um número inteiro positivo, representando o valor a ser obtido utilizando as notas.

Saídas: (dados a serem armazenados no arquivo-texto "notas.txt")

1. Número de notas de 100.
2. Número de notas de 50.
3. Número de notas de 20.
4. Número de notas de 10.
5. Número de notas de 5.
6. Número de notas de 2.
7. Número de notas de 1

O código a seguir possui erros de lógica. Corrija-os para que o programa faça o que se pede.

```
entrada = int(input("Digite o valor inicial"))
cem = entrada//100
cinquenta = entrada // 50
vinte = entrada / 20
dez = entrada / 10
cinco = entrada % 5
dois = entrada % 2
um = entrada

saida=[cem, cinquenta, vinte, dez, cinco, dois, um]
with open ("notas.txt") as arq:
    for i in saida:
        arq.write(i+"\n")
```

Exemplo de entrada	Exemplo de saída
36	0 0 1 1 1 0 1
267	2 1 0 1 1 1 1 0

8) Faça um programa que receba um número inteiro  $n$  e imprima uma pirâmide contendo  $n$  linhas e em cada uma deve conter uma sequência numérica até o número que represente o valor da linha.

O programa deve ter uma subrotina que recebe o valor de  $n$  e imprime (grava) a pirâmide.

Entradas:

1. Número inteiro informando a quantidade de linhas

Saídas (a pirâmide em um arquivo-texto):

1.  $n$  linhas contendo cada uma uma sequência numérica correspondente ao valor da linha.

Exemplo de Entrada:

5

Exemplo de Saída (conteúdo do arquivo-texto):

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

9) Faça um programa para gravar uma lista de palavras em um arquivo-texto. O programa deve ler enquanto não for digitada a palavra "fim". A palavra fim não deve ser armazenada no arquivo. Não é permitido o uso de listas. Assim que uma palavra válida for digitada, ela deve ser armazenada no arquivo.

O programa deve imprimir na tela quantas palavras válidas foram lidas. O nome do arquivo-texto deverá ser informado pelo usuário.

10) Modifique o programa anterior para gravar as palavras em um arquivo binário.

11) Faça um programa que lê o arquivo binário da questão anterior e imprime na tela as palavras.

**Extra.** Para produzir seus brinquedos, Papai Noel recebe o auxílio de seus queridos e fiéis gnomos. Para não sobrecarregar seus queridos auxiliares, a fábrica funciona apenas das 6h00 às 22h00, com cada gnomo trabalhando no máximo doze horas diárias, sendo que a quantidade de horas trabalhadas por um gnomo varia entre os dias da semana. Para controlar isso, ele utiliza um registro contendo o total de horas trabalhadas durante os 7 dias da semana e o nome do gnomo. Com as contratações temporárias de final de ano, Papai Noel já não sabe mais quantos gnomos estão trabalhando atualmente.

Faça um programa que **leia os registros de trabalho de um arquivo texto** e armazene as informações em uma lista.

Faça um subprograma que recebe a lista de dados e imprime o nome dos gnomos que trabalharam mais que 12h diárias durante a semana. Caso nenhum gnomo tenha trabalhado mais que 12 horas diárias, imprima "ninguém".

Faça um subprograma que imprime a lista de gnomos.

Faça um subprograma que retorna a posição no arranjo (lista) do gnomo que menos trabalhou durante a semana.

Faça um subprograma que recebe como um nome de gnomo para procurar na lista. O subprograma deve retornar a posição do gnomo na lista. Caso não encontrado, retornar -1.

OBS: Considere que os nomes dos gnomos não tem espaços e que dois gnomos não podem trabalhar a mesma quantidade de horas.

É obrigatório o uso de **dicionários** para representar dados relativos aos gnomos.

#### **Exemplo de Entrada (conteúdo do arquivo):**

82 Trabalhador  
65 Capaz  
90 Marchalenta  
71 Dorminhoco  
54 Tranquilo  
85 Agitado  
89 Sossego  
75 Astuto

#### **Exemplo de Saída:**

Gnomos que trabalharam mais de 12h:  
Marchalenta  
Agitado  
Sossego  
Gnomo que menos trabalhou:  
4