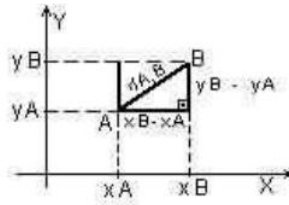


Lista de exercícios em Python

Parte 1: sintaxe básica

1) **[variáveis e operadores]** Desenvolva um programa que calcule a distância entre dois pontos no plano. Os valores dos pontos devem ser informados pelo usuário.



Fórmula:

$$d_{AB} = \sqrt{(X_B - X_A)^2 + (Y_B - Y_A)^2}$$



Utilize as funções do pacote math para a resolução do exercício (import math)

2) **[E/S, laços, comandos de decisão] Jogo da Adivinhação.** Desenvolva um programa em Python que simule um jogo de adivinhação. Primeiramente, o programa deverá sortear um número entre 0 e 100. Após o sorteio, inicia-se o jogo e o jogador deverá tentar adivinhar o número sorteado. A cada tentativa, o jogo deverá informar se o “chute” do jogador foi maior ou menor do que o número a ser adivinhado. O jogo termina após vinte tentativas erradas ou quando o jogador acertar o número sorteado e, nesse caso, informe a quantidades de tentativas que foram necessárias. Após o término do jogo o jogador deverá decidir se deseja ou não iniciar um novo jogo.



from random import *

A função **random()** retorna um float x tal que $0 \leq x < 1$.

A função **uniform(10,20)** retorna um float x tal que $10 \leq x < 20$.

A função **randint(100,1000)** retorna um inteiro x tal que $100 \leq x < 1000$.

A função **randrange(100,1000,2)** retorna um inteiro x tal que $100 \leq x < 1000$ e x é par (ou seja, passo 2)

3) **[Strings, listas, módulos]** Formatação de data - Elaborar uma função que recebe como parâmetros o dia, mês e ano como inteiros. A função deverá retornar a data por extenso conforme o exemplo abaixo:

```
sdata(dia, mes, ano)
```

Exemplo:

```
data=sdata(10,04,2017)
```

```
print data    => imprime: 10 de abril de 2017
```

4) [listas de listas, módulos] A tabela de frequência é um recurso estatístico utilizado para classificar e representar um conjunto de valores. É formada por **n** informações contendo o **limite inferior**, o **limite superior** e a **quantidade de elementos** do vetor que são maiores ou iguais ao limite inferior e menores que o limite superior. A diferença entre o limite superior e o limite inferior de cada registro, também chamada de **amplitude**, é calculada da seguinte forma: $\text{RoundUp}((\text{maior_elemento} - \text{menor_elemento})/n)$. O valor de **n** é calculado pela raiz quadrada da quantidade de elementos do vetor. Dado um vetor de números inteiros ordenados, faça uma função que ao **receber este vetor** e seu **tamanho lógico**, gere e **retorne** uma matriz dinâmica de frequência. **RoundUp** significa arredondamento para cima (não é uma função da linguagem Python), mas pode ser representada pela função **math.ceil**.

Exemplo de tabela de frequência:

vetor : 2,3,4,6,7,9,10,15,17,18,20,22,25,29,30,31,33,36,37,39,50,53,56,80,96

n: $\text{sqrt}(25) = 5$

amplitude: $(96-2)/5 = 18,8 = 19$

Limite inferior	Limite superior	Qtde Elementos
2	21	11
21	40	9
40	59	3
59	78	0
78	97	2

5) [arquivos, módulos] Converta o arquivo cidades.txt disponível no Aprender em um arquivo html 5 de forma que cada cidade apareça em um parágrafo, inserindo um título a cada troca de iniciais do nome. Teste no navegador:

Adamantina
Adolfo
Aguai
Aguas da Prata
Aguas de Lindoia
Aguas de Santa Barbara
Aguas de Sao Pedro
Agudos
Alambari
Alfredo Marcondes
Altair
Altinopolis
Alto Alegre
...

➔

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Cidades do Estado de São
Paulo</title>
  </head>
  <body>
    <h1>A</h1>
    <p> Adamantina </p>
    <p> Adolfo </p>
    <p> Aguai </p>
    <p> Aguas da Prata</p>
    <p> Aguas de Lindoia</p>
    ...
    <h1>B</h1>
    <p>Bady Bassitt </p>
    ...
  </body>
</html>
```



você pode executar um programa externo utilizando o comando `os.system("cidades.html");`
`import os`

Parte 2: POO

Desenvolva uma classe em Python com o objetivo de auxiliar no processo de votação e apuração em eleições. A classe deverá manter uma coleção de candidatos, possibilitar a votação, registrar o tipo de eleição (presidente, diretório acadêmico, prefeito, diretor de clube, etc) e emitir o resultado da apuração. Para isso crie a seguinte estrutura da classe:

- Um construtor que recebe a quantidade de candidatos e o tipo de eleição;
- Um construtor que recebe somente a quantidade de candidatos
- Métodos gets e sets para o tipo de eleição
- Um método que permite adicionar individualmente os candidatos: adicionaCandidato (Pessoa). Deve retornar um booleano.
- Um método que retorna na forma de uma String, a lista de candidatos adicionados. Utilize '\n' para separar cada nome.
- Um método que permite votar, o qual recebe o número do candidato e assim contabilizar um voto a ele.
- Um método que emite o boletim de urna em arquivo, ou seja um arquivo com a quantidade e o percentual de votos para cada candidato. Por exemplo:

```
****Eleição para o DA****
Candidato 10: Pelé 7 votos (70%)
Candidato 33: Maradona 0 votos (0%)
Candidato 14: Ronaldo 3 votos (30%)
```

Os candidatos devem ser representados por objetos da classe **Pessoa**, com os atributos número e nome ou pela classe **Candidato**, herança de Pessoa que além de número e nome, possui o atributo partido.

```
#exemplo de uso
urna=Urna(3,"Eleição para o DA")
urna.adicCandidato(Pessoa(10,"Pelé"));
urna.adicCandidato(Pessoa(33,"Maradona"));
urna.adicCandidato(Pessoa(14,"Ronaldo"));
...
print(urna.listaCandidatos());
urna.votar(10);
urna.votar(10);
urna.votar(14);
...
```