ATIVIDADE PRÁTICA

1. OBJETIVO

Desenvolver os algoritmos colocados no item 5, ao final deste documento, em **linguagem de programação Python**.

2. MATERIAL UTILIZADO

A Atividade Prática de Lógica de Programação e Algoritmos será realizada com a utilização do software **Pycharm** ou do ambiente de desenvolvimento web **Google Colab**. Orientações para instalação e uso do software estão disponíveis em vídeo na ROTA DE APRENDIZAGEM (AULA 7 – Vídeos complementares).

IMPORTANTE: caso o aluno tenha familiaridade com outro ambiente de desenvolvimento em Python, é possível utilizá-lo. Porém, nossas ferramentas oficiais e que teremos tutorial de instalação são o Pycharm e o Google Colab, portanto fica a cargo do aluno saber utilizar as outras ferramentas.

3. ORIENTAÇÕES GERAIS

- O aluno deverá entregar um **ARQUIVO ÚNICO NO FORMATO PDF OU DOCX** no AVA no ícone **Trabalhos**;
- Submeter o trabalho em múltiplos arquivos separados, ou em formatos diferentes dos impostos acima, acarretará em nota zero;
- Esta atividade é para ser realizada com consulta e pesquisa. Portanto, não basta somente estudar o material da rota de aprendizagem. Leia também os livros bases, materiais complementares e procure seu tutor para tirar dúvidas diretamente via Canal de Tutoria;
- Para cada exercício, coloque no seu relatório o ENUNCIADO do mesmo e coloque como resposta o seu código COMPLETO. Deste a primeira até a última linha que você digitou;
- Coloque no seu código COMENTÁRIOS explicando COM SUAS PALAVRAS o que ele faz (veja o exemplo do documento modelo);
- Tanto no Pycharm quanto no Google colab, para inserir os códigos nas respostas, faça um **CTRL+C/CTRL+V** do código criado por você na ferramenta, e cole-o no documento do Word. Assim, o código já virá **colorido**, organizado e indentado, facilitando a correção da sua atividade;

- No AVA existe um modelo em WORD para você utilizar. Se você optar por submeter o seu arquivo em PDF, basta apertar em salvar como PDF no Word;
- Além do seu algoritmo, você deverá colocar uma captura de tela do seu código funcionando. Capture o terminal mostrando o seu código funcionando e imprimindo os dados solicitados na tela (veja o exemplo do modelo).

- **CUIDADO!**
  - ✓ **Em programação, não existem dois códigos exatamente iguais. Cada programador organiza seu código de uma forma diferente, declara variáveis com nomes diferentes, faz comentários diferentes, gera mensagens aos usuários distintas, etc. Por este motivo, e como a atividade é INDIVIDUAL, não serão aceitos dois algoritmos idênticos entre alunos (ou iguais à Internet). Caso o corretor observe respostas iguais, elas serão consideradas como PLÁGIO e será atribuída a NOTA ZERO na questão;**

4. COMO SE DARÁ A CORREÇÃO DA ATIVIDADE?

Como temos 4 questões. Seus pesos são de 25% no total da atividade cada um;

Para que você ganhe nota máxima em cada exercício, você precisará cumprir os três requisitos básicos explicados nas ORIENTAÇÕES GERAIS:

- Apresentar seu algoritmo completo, indentado e organizado;
- Explicar seu código através de comentários;
- Colocar uma IMAGEM com o terminal rodando e mostrando o que cada exercício pede.

No modelo de relatório da disciplina você encontrará um exemplo de exercício para um melhor entendimento. Caso você desenvolva seu código corretamente e funcional, porém não faça os comentários nem coloque uma imagem dele funcionando no terminal, terá sua nota severamente prejudicada.

5. EXERCÍCIOS

Resolva os algoritmos abaixo em Python seguindo todas as instruções listadas neste documento.

Exercício 1:

Escreva um programa que leia o nome de um lutador e seu peso. Em seguida, informe a categoria a que pertence o lutador, conforme a Tabela a seguir (note que a tabela foi criada para efeito deste exercício e não condiz com qualquer categoria de luta). A saída do programa deve exibir na tela uma frase com o padrão descrito a seguir:

Nome fornecido: Pepe Jordão
Peso fornecido: 73.4
Frase a ser exibida: O lutador Pepe Jordão pesa 73,4 kg e se enquadra na categoria Ligeiro

Peso	Categoria
Menor que 65 kg	Pena
Maior ou igual a 65 kg e menor que 72 kg	Leve
Maior ou igual a 72 kg e menor que 79 kg	Ligeiro
Maior ou igual a 79 kg e menor que 86 kg	Meio-médio
Maior ou igual a 86 kg e menor que 93 kg	Médio
Maior ou igual a 93 kg e menor que 100 kg	Meio-pesado
Maior ou igual a 100 kg	Pesado

Todos os dados devem ser lidos do teclado, sendo que o nome do lutador é string e o peso é um número real. Não é necessário armazenar os dados em uma estrutura de dados, basta imprimir na tela.

Coloque todo o seu programa dentro de um laço de repetição e faça-o se encerrar quando uma determinada condição for satisfeita. A condição fica a seu critério.

Imprima na tela um teste do seu programa utilizando o seu nome e os dois últimos dígitos do seu RU para o peso.

Exercício 2:

Escreva um programa que receba como parâmetro de entrada um número inteiro de 5 dígitos no intervalo fechado [10000, 30000] que represente códigos de produtos vendidos em uma loja. Crie uma função para validar os dados de entrada, obrigando o usuário a respeitar o intervalo e o tipo de dado (inteiro).

Crie uma função que calcule e retorne o dígito verificador do código, utilizando a regra de cálculo explicada a seguir. Por exemplo, considere o código 21853, em que cada dígito é multiplicado por um peso começando em 2, os valores obtidos são somados, e do total obtido calcula-se o resto de sua divisão por 7.

Dígito	2	1	8	5	3	
Peso	2	3	4	5	6	
Multiplicação	4	3	32	25	18	Soma todos = 82
						Resto de 82 por 7 = 5

Retorne na função o valor do dígito verificador calculado e imprima na tela o código do produto digitado e seu dígito verificador separado por hífen, como: 21853-5.

Imprima na tela um teste do seu programa utilizando como código os 5 primeiros dígitos do seu RU. Se seu RU tiver menos de 5 dígitos, complete com zeros. Se seu RU cair fora do intervalo especificado, realize o teste mesmo assim.

Exercício 3:

Considere o seguinte conjunto de dados: Nome + (N1, N2, N3, N4). Nome representa o nome de um aluno e deve ser usado como chave. Já N1, N2, N3, N4 representam as notas de provas desse aluno. Utilize uma estrutura de dicionário com listas para resolver este exercício.

Escreva um programa que leia os dados de N alunos e apresente na tela se foram aprovados ou reprovados. O critério que garante a aprovação é que a média aritmética das 4 notas seja maior ou igual 7,0. O valor de N é a quantidade de alunos, e esse valor deve ser lido do teclado no começo do programa. Faça um laço de repetição para a leitura destes N alunos. As notas devem ser exibidas ao final do programa com uma casa decimal de precisão.

Sugestão de apresentação dos resultados:

```
Notas dos alunos
- - - - - - - - - -
Cláudio dos Santos    6.5  3.5  5.0  6.5  5.4  Reprovado
Eduardo Falco         7.5  6.5  7.0  6.5  6.9  Aprovado
Lígia de Oliveira     8.5  7.5  8.0  7.0  7.9  Aprovado
```

Imprima na tela um teste do seu programa usando como primeiro cadastro o seu nome, e como nota os 4 primeiros dígitos do seu RU.

Exercício 4:

Leia e armazene em um dicionário o nome, a idade e o número do telefone de seus contatos, sendo que a chave deve ser o nome. Ao digitar uma string vazia para o nome, o programa interrompe a leitura e se encerra.

Apresente na tela os dados lidos em ordem alfabética pelo nome dos contatos. Uma possível solução de ordenar alfabeticamente é usar o método *sort.*

Em seguida, armazene os contatos em outros dois dicionários, utilizando como critério a idade: menores de 18 anos em um e os maiores em outro dicionário, eliminando o original. Apresente na tela os dois dicionários resultantes da separação.

Imprima na tela um teste do seu programa usando como primeiro cadastro o seu nome, como telefone o seu RU, e como idade os dois últimos dígitos do seu RU.