

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 aluno e sua nota final. Em seguida, informe o conceito conforme a tabela abaixo. A saída do programa deve exibir na tela uma frase com o padrão descrito a seguir:

Nome do aluno: Fábio José
Nota final: 73.4

Frase a ser exibida: O aluno Fabio José tirou nota 3.5 e se enquadra no conceito D

Nota	Conceito
De 0,0 a 2,9	E
De 3,0 a 4,9	D
De 5,0 a 6,9	C
De 7 a 8,9	B
De 9,0 a 10	A

Todos os dados devem ser lidos do teclado, sendo que o nome do aluno é uma string e a nota final 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, como por exemplo, encerrar o programa ao digitar a palavra sair, ou então uma nota inválida.

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

```
Inserir dados? 0 - Não      1-Sim  1
Nome do aluno: Ana
Nota final: 9
O aluno Ana tirou a nota 9.0 e se enquadra no conceito A
Inserir dados? 0 - Não      1-Sim
```

Conhecimento mínimo necessário para resolver este exercício: até a AULA 4.

Ficou com dúvida no seu exercício? Está com dificuldades em avançar na solução?

Encaminhe seu código na tutoria em modo texto com sua pergunta para que possamos auxiliar você.

Exercício 2:

Faça uma **função** que receba o nome e sobrenome de uma pessoa e retorne a primeira letra de seu nome e seu sobrenome concatenando com a string @algoritmos.com.br. No algoritmo principal deverá ser apresentada a mensagem ao usuário contendo seu nome completo e seu email.

Imprima na tela um teste do seu programa utilizando o seu nome e sobrenome concatenado com os dois últimos dígitos do seu RU.

Sra Luciane Kanashiro, seu email é ikanashiro16@algoritmos.com.br

Conhecimento mínimo necessário para resolver este exercício: até a AULA 5.

Ficou com dúvida no seu exercício? Está com dificuldades em avançar na solução?

Encaminhe seu código na tutoria em modo texto com sua pergunta para que possamos auxiliar você.

Exercício 3:

Um canal de jogos do youtube está fazendo um sorteio para angariar doações para pessoas em situação de vulnerabilidade social. A cada 10,00 doado o nome da pessoa é inserido em uma lista de sorteio, por exemplo:

Ruth doou 20,00;

Maria doou 30,00;

Fernando doou 50,00;

A lista de sorteio estará com os valores:

```
listaSorteio = ['Ruth', 'Ruth', 'Maria', 'Maria', 'Maria', 'Fernando', 'Fernando', 'Fernando', 'Fernando', 'Fernando']
```

Implemente um programa para cadastrar o nome das pessoas que doaram. O programa deve embaralhar a lista, sortear o ganhador e imprimir o seu nome.

Imprima na tela um teste do seu programa utilizando como primeiro doador o seu nome e os dois últimos dígitos do seu RU para o valor doado. Não se esqueça de imprimir também a lista de sorteio.

Dica do Professor:

Utilize o método `random.shuffle` para embaralhar a lista;

Utilize método `random.choice` para sortear aleatoriamente um elemento.

Abaixo, um exemplo de como utilizar os métodos para embaralhar e sortear:

```
import random
lista = [1,2,3,4,5]
random.shuffle(lista)
sorteado = random.choice(lista)
print(sorteado)
```

Conhecimento mínimo necessário para resolver este exercício: até a AULA 6.

Ficou com dúvida no seu exercício? Está com dificuldades em avançar na solução?

Encaminhe seu código na tutoria em modo texto com sua pergunta para que possamos auxiliar você.

Profa. Luciane Kanashiro, Me.

Exercício 4:

Considere a tabela a seguir referente a produtos armazenados em um depósito, em que são considerados o estoque atual de cada produto e o estoque mínimo necessário.

Código	Estoque	Mínimo
1	35	20
5	75	50
2	43	45
3	26	18
20	35	20

Armazene as informações acima em uma estrutura de lista com dicionário, substituindo a primeira linha com os dados: no campo código coloque o primeiro dígito do seu RU, no estoque os dois dígitos seguintes de seu RU, e no campo mínimo os dois últimos dígitos do seu RU.

Por exemplo, tendo o RU: **1****23****456**

Código	Estoque	Mínimo
1	23	56
5	75	50
2	43	45
3	26	18
20	35	20

As informações devem ser inseridas no dicionário via teclado. Ao digitar o código 0 (zero), o programa interrompe a leitura e se encerra. Ordene os produtos em ordem crescente de **código**. Imprima na tela um teste do seu programa usando como primeiro cadastro o primeiro dígito do seu RU, como estoque os dois dígitos seguintes de seu RU, e como mínimo os dois últimos dígitos do seu RU.

Dica: A sua lista deve conter o dicionário da seguinte forma:

```
[{'codigo': 1, 'estoque': 23, 'minimo': 56}, {'codigo': 5, 'estoque': 75, 'minimo': 50}, ...,  
{ 'codigo': 20, 'estoque': 35, 'minimo': 20}]
```

Para criar dicionário e lista:

```
dicionario = { }
```

```
lista = [ ]
```

Para ordenar: utilize o método sorted, ordenando pela chave codigo

```
listaOrdenada = sorted(lista, key=itemgetter('codigo'))
```

Não se esqueça de importar o itemgetter no início do seu programa.

```
from operator import itemgetter
```

Ou utilize o lambda:

```
listaOrdenada = sorted(lista, key=lambda item: item['codigo'])
```

Conhecimento mínimo necessário para resolver este exercício: até a AULA 6.

Ficou com dúvida no seu exercício? Está com dificuldades em avançar na solução?

Encaminhe seu código na tutoria em modo texto com sua pergunta para que possamos auxiliar você.