



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
DEPARTAMENTO DE COMPUTAÇÃO
IC592 – Linguagem de Programação I

TR1 - Trabalho 1 de Avaliação 23/02/2022

Prof. Claver Pari Soto

Data de entrega: 09/03/2022 até 23:59 hrs

Os seguintes problemas devem ser resolvidos e enviados ao professor via e-mail.

Critérios de avaliação:

Relatório grupal (Bem formatado, com as descrições dos objetivos alcançados, relato das dificuldades, das soluções, etc)	Até 3,0
Código apresentado (Solução alcançada)	Até 3,0
Total:	Até 6,0

Devem ser enviados ao professor:

1. Um documento em **.pdf** com o nome de arquivo seguindo o seguinte formato:

IC501_TR1_<NomeSobrenome1>_<NomeSobrenome2>_<...>.pdf

Por exemplo, se o grupo tem 2 integrantes: Ana Mendes e Igor Nunes, e sua turma é T05 o arquivo deve ter o nome:

IC501_TR1_AnaMendes_IgorNunes.pdf

- Neste documento, cada problema resolvido deve ter seu título “Problema 1”, “Problema 2”, etc que corresponda ao problema proposto.
- Se um problema não foi resolvido, deve constar: “Este problema não foi resolvido”
- Cada problema resolvido deve conter a colagem simples do código do programa. Não precisa ter número de linha.
- Além da colagem do código, cada solução deve conter as observações, argumentações, etc. pertinentes a cada problema. Lembre, que o grau de detalhamento desses apontamentos é muito importante

2. Um documento zipado .zip com o nome de arquivo seguindo o seguinte formato:

IC501_TR1_<NomeSobrenome1>_<NomeSobrenome2>_<...>.zip

Por exemplo,

IC501_TR1_AnaMendes_IgorNunes.zip

- Neste documento, devem estar zipados todos os códigos fonte em Python, “problema_1_resolvido.py”, “problema_2_resolvido.py”, etc.
- A primeira linha de cada arquivo fonte (.ipynb) deve ser de comentário, por exemplo:
Elaborado por: Ana Mendes e IgorNunes

Bom trabalho!

Problema 1 (1,0 pontos):

Suponha que você queira depositar uma certa quantia em uma aplicação bancária, e deixá-la obtendo juros (com a **taxa_de_juros**) pelos próximos **nun_anos** anos. Ao final desses **num_anos**, você gostaria de ter um **valor_futuro** na conta. Quanto você precisa depositar hoje (**valor_presente**) para que isso aconteça? Você pode usar a seguinte fórmula para descobrir:

$$P = \frac{F}{(1 + r)^n}$$

Os termos da fórmula são os seguintes:

- **P** é o valor presente ou a quantia que você precisa depositar hoje.
- **F** é o valor futuro que você deseja na conta.
- **r** é a taxa de juros anual.
- **n** é o número de anos que você planeja deixar o dinheiro ficar na conta.

Escreva um programa em Python que realize o cálculo para encontrar a resposta ao problema, e para casos similares onde somente os valores das variáveis mudem.

Aqui está um algoritmo que pode ser usado:

1. O programa deve perguntar ao usuário o valor futuro desejado.
2. O programa deve perguntar ao usuário pelo valor da taxa de juros anual.
3. O programa deve perguntar ao usuário pelo número de anos que o dinheiro permanecerá na conta.
4. Calcular o valor que deverá ser depositado.
5. Exibir o resultado que se obteve no cálculo na etapa 4.

Problema 2 (1,0 pontos):

Escreva um programa que preveja o tamanho aproximado de uma população de mosquitos de fruta em um ambiente fechado. O aplicativo deve perguntar ao usuário pelo número inicial de mosquitos (Dia 0), pelo aumento médio diário da população (como uma porcentagem), e pelo número de dias que os mosquitos terão para se reproduzir.

Por exemplo, suponha que o usuário insira os seguintes valores:

Número inicial de mosquitos:

Aumento diário médio (%):

Número de dias para reprodução:

O programa deve exibir uma tabela similar à seguinte:

```
Número inicial de mosquitos: 50
Aumento diário médio (%): 30
Número de dias para reprodução: 5
```

Dia	População
0	50.0
1	65.0
2	84.5
3	109.8
4	142.8
5	185.6

Problema 3 (1,0 pontos):

Imagine uma clínica médica com 5 consultórios. Cada dia é definido um número máximo de consultas igual para cada consultório (**maximo_consultas**). Sabe-se que diariamente todos os consultórios têm pelo menos 2 pacientes atendidos (pre-agendados).

Faça um programa que pergunte ao usuário qual será o valor de **maximo_consultas**, logo o programa deve gerar 5 números aleatórios que representem o número de pacientes atendidos em cada consultório. Para finalizar, o programa deve informar ao usuário o número médio de pacientes atendidos nos 5 consultórios da clínica.

Dica:

Para a geração de um número aleatório, você pode usar a função **randrange()** do módulo **random** no Python, que por exemplo, no seguinte código imprime na tela um número aleatório inteiro não negativo menor que 4.

```
import random
print("número aleatório = ", random.randrange(4))
```

Exemplos de execução (meramente ilustrativos):

Ingresse o número máximo de consultas:
12
média de consultas = 8.8
Ingresse o número máximo de consultas:
25
média de consultas = 17.0

Problema 4 (1,0 pontos):

Com o seguinte código pretende-se preencher a matrix 5x4 com os valores ao lado. Descubra e coloque o código necessário no comando de atribuição.

	0	1	2	3	4
0	1	3	5	7	9
1	2	4	6	8	10
2	3	5	7	9	11
3	4	6	8	10	12

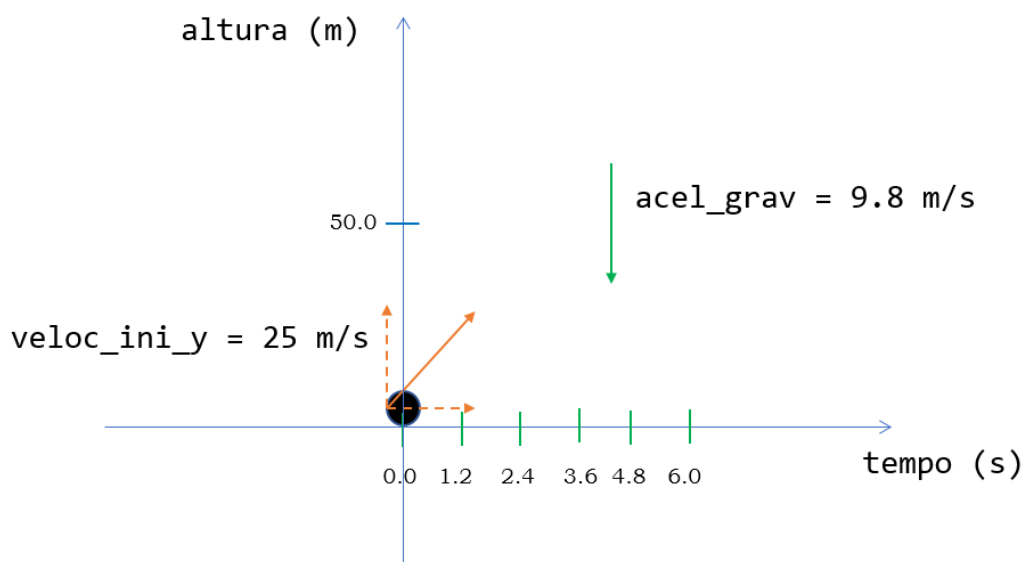
matriz5x4.py:

```
1  tabela = [[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0]]
2  for i in range(4):
3      for j in range(5):
4          tabela[i][j] = _____
5
6  for i in range(4):
7      for j in range(5):
8          print(f"{tabela[i][j]:>3}",end=' ')
9      print()
```

Problema 5 (2,0 pontos):

Usando somente a fórmula da física $h = v_{0y}t - \frac{gt^2}{2}$

- a) Escreva um programa em Python que calcule e imprima as alturas respectivas para cada um dos valores de tempo discretizado indicado na figura abaixo, ou seja, para os 6 pontos: **0.0, 1.2, 2.4, 3.6, 4.8 e 6.0**



- b) Faça mudanças no programa do item (a) para que identifique a altura máxima entre essas 6 alturas calculadas.
- c) Repare que podemos parametrizar esses 6 pontos discretizados de tempo com as seguintes definições:

```
primeiro_ponto = 0.0
último_ponto   = 6.0
número_de_pontos = 6
```

Pense em um algoritmo para calcular automaticamente esses 6 valores mencionados, utilizando as variáveis definidas acima e algumas manipulações algébricas. Não use funções prontas da internet.

Verifique se a resposta coincide com o esperado (item (a))

- d) Agora teste com

```
primeiro_ponto = 0.0
último_ponto   = 6.0
número_de_pontos = 10
```

- e) Repita (a) e (b) considerando este novo conjunto de 10 valores.
- f) Repita (a) e (b) e (d) considerando **número_de_pontos = 1000**