



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina Fundamentos de Programação**  
**Professores: Dante Corbucci Filho e Luís Felipe Ignácio Cunha**  
**AD2 – 1º semestre de 2021**

---

**IMPORTANTE:** As respostas (programas) deverão ser entregues pela plataforma em um arquivo ZIP contendo todos os arquivos de código fonte (extensão “.py”) necessários para que os programas sejam testados. Respostas entregues fora do formato especificado, por exemplo, em arquivos com extensão “.pdf”, “.doc” ou outras, não serão corrigidas.

- Serão aceitos apenas soluções escritas na linguagem Python 3. Programas com erro de interpretação não serão corrigidos. Evite problemas utilizando tanto a versão da linguagem de programação (Python 3.X) quanto a IDE (PyCharm) indicadas na Aula 1.
- Quando o enunciado de uma questão inclui especificação de formato de entrada e saída, tal especificação deve ser seguida à risca pelo programa entregue. Atender ao enunciado faz parte da avaliação e da composição da nota final.
- Os exemplos fornecidos nos enunciados das questões correspondem a casos específicos apontados para fins de ilustração e não correspondem ao universo completo de entradas possíveis especificado no enunciado. Os programas entregues devem ser elaborados considerando qualquer caso que siga a especificação e não apenas os exemplos dados. Essa é a prática adotada tanto na elaboração das listas exercícios desta disciplina quanto no mercado de trabalho.
- Faça uso de boas práticas de programação, em especial, na escolha de identificadores de variáveis, subprogramas e comentários no código.
- As respostas deverão ser entregues via atividade específica na Plataforma antes da data final de entrega estabelecida no calendário de entrega de ADs. Não serão aceitas entregas tardias ou substituição de respostas após término do prazo.
- As ADs são um mecanismo de avaliação individual. As soluções podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual. Respostas plagiadas não serão corrigidas.

---

**Boa Avaliação!**

### 1ª Questão (2,5 pontos)

Faça um programa, contendo subprogramas, que leia da entrada padrão o nome de um arquivo do tipo texto contendo informações sobre a vacinação da Covid. Mostre inicialmente seu conteúdo. Cada linha possui as seguintes três informações:

nome#vacina#dose

Onde vacina pode ser uma das seis opções a seguir: “Oxford”, “Coronavac”, “Pfizer”, “Sputnik”, “Covax” ou “Jansen”. Produza sete arquivos, sendo que o primeiro deve conter em cada linha os nomes dos cidadãos que receberam duas doses, no formato nome#vacina. Os outros seis arquivos devem conter apenas os nomes dos cidadãos, um por linha, que receberam apenas uma vacina daquele tipo. Ao final, mostre conteúdo dos sete arquivos produzidos. Suponha que o arquivo original seja consistente e que ninguém tenha tomado uma dose de uma vacina e uma dose de outra. Além disso, ninguém toma a segunda sem ter tomado a primeira vacina.

Restrição: Está proibido o uso de listas para manter nomes de pessoas e vacinas, etc. Suas eventuais listas devem possuir no máximo sete elementos. Caso esta restrição não seja respeitada, sua solução perderá 50% do valor da questão. Isto é, toda a massa de informações deve residir em arquivo do tipo texto.

#### Teste:

Entrada	Saída
Teste1	Listagem das Aplicações de Vacina Teste1: Edson Arantes do Nascimento#Pfizer#1 Gal Costa#Oxford#1 Milton Nascimento#Pfizer#1 Gilberto Gil#Coronavac#1 Gal Costa#Oxford#2 Rainha Elizabeth#Oxford#1 Chico Buarque#Coronavac#1 Gilberto Gil#Coronavac#2 Paul McCartney#Oxford#1 Edson Arantes do Nascimento#Pfizer#2 Bruce Lee#Covax#1 Rainha Elizabeth#Oxford#2 Magic Johnson#Jansen#1 Garry Kasparov#Sputnik#1 Vacinados com duas doses em Teste1: Gal Costa#Oxford Gilberto Gil#Coronavac Edson Arantes do Nascimento#Pfizer Rainha Elizabeth#Oxford  Vacinados com uma dose da Coronavac em Teste1: Chico Buarque  Vacinados com uma dose da Oxford em Teste1: Paul McCartney  Vacinados com uma dose da Pfizer em Teste1:

	Milton Nascimento  Vacinados com uma dose da Sputnik em Teste1: Garry Kasparov  Vacinados com uma dose da Covax em Teste1: Bruce Lee  Vacinados com uma dose da Jansen em Teste1: Magic Johnson
--	--

## 2ª Questão (2,5 pontos)

Faça um programa, contendo subprogramas, que leia da entrada padrão o nome de um arquivo do tipo texto contendo em cada linha o nome de um apostador e oito números diferentes escolhidos por ele no intervalo 0 a 99. Suponha que cada item seja separado pelo caractere “#”. Ou seja:

nome do apostador#n1#n2#n3#n4#n5#n6#n7#n8

Mostre seu conteúdo na saída padrão. Em seguida, leia da entrada padrão uma linha contendo o resultado do sorteio de oito números diferentes, separador por espaço em branco. Ou seja:

res1 res2 res3 res4 res5 res6 res7 res8

Suponha que no arquivo ninguém possa fazer mais que uma aposta.

Processe o arquivo e produza um dicionário (tipo “dict”) onde a chave é o número de acertos 8, 7, 6, 5, 4 ou 3, e o valor é um conjunto (tipo “set”) dos nomes de todos os apostadores com aquele total de acertos.

Caso nenhuma aposta esteja contida no arquivo, escreva a mensagem “Nenhuma Aposta!!!”. Caso contrário, escreva na saída padrão o total de apostas no arquivo e de acertadores de cada grupo, iniciando com 8 acertos e finalizando com 3 acertos e seus respectivos nomes, ordenados alfabeticamente de forma crescente. Caso ninguém tenha acertado 3 ou mais números, escreva a mensagem “ACUMULOU TUDO”.

## Teste

Entrada	Saída
Teste2 1 2 3 4 5 6 7 53	Conteúdo do Arquivo de Apostas Teste2: Juca Kfourir#10#15#29#33#82#7#1#53 Gustavo Kuerten#3#5#99#88#40#97#1#2 Novak Djokovic#6#12#53#88#1#2#3#4 Rafael Nadal#1#2#3#4#71#72#73#63 ---- Fim do Arquivo de Apostas ----

	Total de Apostas: 4 Ninguém Acertou 8 Números!!! Ninguém Acertou 7 Números!!! Foi(ram) 1 Ganhador(es) com 6 Acertos: Novak Djokovic Ninguém Acertou 5 Números!!! Foi(ram) 2 Ganhador(es) com 4 Acertos: Gustavo Kuerten Rafael Nadal Foi(ram) 1 Ganhador(es) com 3 Acertos: Juca Kfourti
--	--

### 3ª Questão (3,0 pontos)

Escreva um programa que faça análises sobre consumos de carros movidos a álcool e a gasolina. Seu programa deve avaliar com carro mais econômico no ano desejado, de acordo com a quilometragem desejada do carro andar na cidade e na estrada. Segue abaixo todos os detalhes a serem feitos.

A entrada é dada por: (1) um arquivo binário, (2) um arquivo texto e (3) três inteiros, com as seguintes exigências:

(1) Um arquivo binário:

O arquivo binário de nome “carros.bin” é composto por uma sequência de registros formados por duas strings, dois pares de dois pontos flutuantes de precisão dupla e um ponto flutuante de precisão dupla:

Duas strings, onde a primeira ocupa 4 bytes e a segunda ocupa 16 bytes. A primeira string se refere ao ano do carro e a segunda ao modelo do carro;

O primeiro par de pontos flutuantes associa a quantidade de KM por L rodados pelo carro à álcool e a gasolina, respectivamente, na cidade;

O segundo par de pontos flutuantes associa a quantidade de KM por L rodados pelo carro à álcool e a gasolina, respectivamente, na estrada;

O terceiro ponto flutuante associa a quantos litros que há no tanque do carro.

(2) Um arquivo texto:

O arquivo texto de nome “precos\_combustiveis.txt” é composto por um par de pontos flutuantes de precisões tripla com os preços do álcool e da gasolina, respectivamente.

(3) Três inteiros:

O primeiro inteiro será o ano referente a consulta a ser feita;

O segundo inteiro será a quilometragem do carro na cidade a ser avaliada;

O terceiro inteiro será a quilometragem do carro na estrada a ser avaliada.

A saída deverá ser em um arquivo "carro\_mais\_economico.txt" e deve possuir as seguintes informações:

Ano avaliado, o carro mais econômico utilizando somente gasolina e somente álcool no ano desejado, juntamente com o quanto do tanque foi utilizado e o valor gasto para a quilometragem.

#### Exemplo

Entrada	Saída (carro mais economico.txt)
<pre> carros.bin precos_combustiveis.txt 2018 150 250 </pre>	<p>No ano de 2018, os carros mais econômicos para andar 400 km à gasolina e a álcool são Stepway e Kwid, respectivamente.</p> <p>À gasolina, para andar 400 km utilizaremos 0.56 tanques e gastaremos R\$ 170.00.</p> <p>À álcool, para andar 400 km utilizaremos 0.79 tanques e gastaremos R\$ 151.96.</p>
Arquivo (carros.bin)	Arquivo (precos_combustiveis.txt)
<pre> 2017 Duster 7 7 10 11 50 2018 Kwid 7 9 12 13 48 2017 Aircross 7 8 11 12 50 2018 Stepway 8 12 10 14 50 </pre>	<pre> 4.002 6.102 </pre>

**Obs.: O arquivo .bin aparece no exemplo dessa forma para melhor compreensão da questão.**

**Obs.: Os valores apresentados são meramente ilustrativos. Pode não haver nenhum grande reflexo com a realidade.**

#### **4ª Questão (2,0 pontos)**

Escreva um programa que computa os preços de um determinado produto juntamente com o frete do mesmo. A entrada contém dois arquivos e uma string: (1) "frete.bin", que consiste de CEP's, 8 bytes, juntamente do valor do frete do produto; (2) "produto.txt", que consiste das lojas juntamente do valor do produto desejado; (3) string que consiste do CEP onde deseja-se a entrega, que pode estar do formato xx.xxx-xxx, xx.xxx.xxx ou xxxxxxxx.

A saída deve ter na tela a loja que possui o menor preço com o frete e o preço total desse produto. Caso o CEP não exista em frete.bin, então a saída deve ser: "O produto desejado não pode ser entregue neste frete".

### Exemplo

Entrada	Saída
frete.bin produto.txt 00000990	A melhor loja do produto1 para o CEP 00000990 é a loja2, com valor total de R\$ 169.30.
Arquivo (frete.bin)	Arquivo (produto.txt)
00000990 loja1 78.90 00000990 loja2 48.90 00000990 loja3 100.20 12300990 loja1 78.90 12300990 loja2 50.88 12300990 loja3 21.90	produto1  loja1 127.58 loja2 120.40 loja3 90.78