



**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**

**APX2 2º semestre de 2021**

---

### **IMPORTANTE**

**MUITA ATENÇÃO:** Nesta APX há, na próxima página, o TERMO de CONDUTA que todo aluno deve estar ciente e assinar devidamente o documento. Você pode redigi-lo e assinar manuscrito ou assinar digitalmente, e deve fazer upload deste termo juntamente com o arquivo ZIP das suas respostas. Caso o TERMO de CONDUTA não seja devidamente entregue e assinado, a APX1 não será corrigida e a nota da APX1 será zero.

- 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 pela atividade específica na Plataforma antes da data final de entrega estabelecida. Não serão aceitas entregas tardias ou substituição de respostas após término do prazo.
- As APXs 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.



## TERMO DE CONDUTA

Declaro assumir o compromisso de confidencialidade e de sigilo escrito, fotográfico e verbal sobre as questões do exame ou avaliação pessoal que me serão apresentadas, durante o curso desta disciplina. Comprometo-me a não revelar, reproduzir, utilizar ou dar conhecimento, em hipótese alguma, a terceiros, e a não utilizar tais informações para gerar benefício próprio ou de terceiros. Reitero minha ciência de que não poderei fazer cópia manuscrita, registro fotográfico, filmar ou mesmo gravar os enunciados que me são apresentados. Declaro, ainda, estar ciente de que o não cumprimento de tais normas caracteriza infração ética, podendo acarretar punição de acordo com as regras da minha universidade.

Ciente,

Assinatura do Aluno

### Questão 1 – (Valor 5.0 pontos)

Faça um programa em Python contendo subprogramas. Seu programa deve inicialmente ler da entrada padrão o nome de arquivo texto, contendo uma matriz bidimensional de valores inteiros. Cada linha do arquivo texto contém os valores de uma linha de valores inteiros da matriz, separados por espaço(s) em branco. Conforme formato a seguir:

valorLinha1Coluna1 valorLinha1Coluna2 ... valorLinha1ColunaM

...

valorLinhaNColuna1 valorLinhaNColuna2... valorLinhaNColunaM

Por facilidade, suponha que os valores estejam no intervalo 0 a 999.

Carregue o conteúdo do arquivo em uma matriz bidimensional de valores, isto é, em uma lista de lista. Mostre o conteúdo da matriz na saída padrão, conforme testes a seguir. Em seguida, identifique e ordene o conteúdo da linha de maior soma de valores. Para tanto, utilize método de ordenação elaborado por você. Novamente, mostre o conteúdo da matriz na saída padrão, após a eventual ordenação. Finalmente, salve o conteúdo da matriz eventualmente modificada no arquivo texto foi escolhido inicialmente, isto é, destrua todo o conteúdo anterior que estava no arquivo e escreva as linhas da matriz modificada, preservando o formato estabelecido.

### Arquivos Teste:

Conteúdo do Arquivo Delta.txt:	Conteúdo do Arquivo Gama.txt:
1 10 5 7 333 222 11 211 999 99 9 95 8 888 88 100	1 10 5 7 0 22 8 4 2 333 8 11 299 15 10 5 7 248 111 19 1 55 1 10 62 3 123

Teste 1:

Entrada:

Delta.txt

### Saídas Correspondentes:

Conteúdo da Matriz Lida Antes de Ordenar:	Conteúdo da Matriz Após Ordenar Linha de Maior Soma:
1 10 5 7 333 222 11 211 999 99 9 95 8 888 88 100	1 10 5 7 333 222 11 211 9 95 99 999 8 888 88 100

Teste 2:  
Gama.txt

**Saídas Correspondentes:**

Conteúdo da Matriz Lida Antes de Ordenar:	Conteúdo da Matriz Após Ordenar Linha de Maior Soma:
1 10 5 7 0 22 8 4 2 333 8 11 299 15 10 5 7 248 111 19 1 55 1 10 62 3 123	1 10 5 7 0 22 8 4 2 5 7 8 10 11 15 248 299 333 111 19 1 55 1 10 62 3 123

**Questão 2 – (Valor 5.0 pontos)**

O Natal está chegando, com isso, Tungstênio está organizando a lista de presentes para os seus parentes. Tungstênio recebeu uma relação dos presentes que cada um dos seus parentes deseja receber. Seu programa deve responder o total em reais que Tungstênio vai gastar para comprar os presentes para uma pessoa. Seguem abaixo todos os detalhes a serem feitos.

A entrada é dada por: (1) um arquivo binário codificado de um arquivo texto, (2) nome a ser consultado:

(1) Um arquivo binário **(2.0 pontos)**:

Implemente a codificação para a geração de um arquivo .bin a partir de um arquivo .txt que contenha as informações:

nome#parentesco#presente1&valor1;presente2&valor2;...;presentek&valork

O valor de cada presente está escrito no formato com os centavos aparecendo após a vírgula, exemplo: 1800,00.

O arquivo binário de nome “natal.bin” é, portanto, composto por uma sequência de registros formados por 3 strings, tais que:

A primeira string ocupa 20 bytes e se refere ao nome do parente;

A segunda string ocupa 20 bytes e se refere ao parentesco do Tungstênio;

A terceira string ocupa 100 bytes e se refere aos presentes e valores, cada par no formato presente&valor, e os pares são separados por ponto e vírgula (;).

(2) Nome a ser consultado **(3.0 pontos)**:

Ao consultar um nome que esteja na lista de presentes, o programa deve retornar o valor total a ser gasto por Tungstênio, incluindo o parentesco da pessoa e os produtos da lista. Caso o nome da pessoa não esteja na lista, então o programa deve retornar dizendo que esse nome não está na lista. O formato da saída deve ser estritamente tal como segue no teste a seguir:

Entrada	Saída
natal.bin Nome1	Para comprar os presentes para Nome1, que é meu/minha Esposa, gastarei R\$ 5950.0. Os itens da lista para Nome1 são: Luminaria Viagem Celular
natal.bin Nome10	Nome10 não está na lista
Arquivo (natal.bin)	
Nome1#Esposa#Luminaria&150,00;Viagem&4000,00;Celular&1800,00 Nome2#Filho#Celular&1800,00;Roupa&500,00;Celular&1800,00 Nome3#Mãe#Blusa&80,00;Sapato&150,00;Viagem&1500,00;Celular&1800,00 Nome4#Pai#Sapato&150,00	

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

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

**Boa Avaliação!**