



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Fundamentos de Programação

AP2X 1º semestre de 2020

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 pela atividade "Entrega de AP1X" 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.
-

1ª Questão (3,0 pontos)

Faça um programa, contendo subprogramas, que leia da entrada padrão o nome de um arquivo texto, contendo matriz bidimensional de números inteiros. Suponha que cada linha do arquivo contenha todos os números de uma linha da matriz, separados por um espaço em branco. Leia o arquivo e escreva na saída padrão qual é o maior valor a cada linha lida, e a média geral de todos os valores contidos na matriz. Em seguida, escreva todas linhas cuja média de valores esteja acima da média geral.

Exemplo

Entrada Padrão	Saída Padrão
minhaMatriz.txt	Maior da Linha 1: 200
Conteúdo do Arquivo "minhaMatriz.txt"	Maior da Linha 2: 900
	Maior da Linha 3: 130
	Média Geral: 248.2
	Linha(s) com média(s) acima da Média Geral:
100 200 150 80 190	200 900 800 70 500
200 900 800 70 500	
130 130 130 13 130	

Distribuição de Pontos

Entrada – 0,5 pontos; Processamento – 2,0 pontos; Saída – 0,5 pontos.

2ª Questão (4,0 pontos)

Faça um programa, contendo subprogramas, que leia o nome de um arquivo texto contendo todos os celulares das pessoas de uma cidade. Suponha que para cada celular exista um outro arquivo texto contendo a cronologia do posicionamento x e y, tomados com algum intervalo temporal de um dia. Cada linha do arquivo contém um ponto. Calcule a distância percorrida por cada celular, dada pelo somatório de distâncias entre dois pontos consecutivos no arquivo do celular. Escreva na saída padrão todos os celulares que percorreram mais que quinhentos metros. Escreva também a distância média de deslocamento dos celulares da cidade.

Definição

A distância entre dois pontos (x1, y1) e (x2, y2) é dada pela raiz quadrada da soma do quadrado das diferenças (x2-x1) e (y2-y1).

Exemplo

Entrada Padrão	Saída Padrão
paris.txt	434370003 percorreu: 998.044207459363 metros
Conteúdo do Arquivo "paris.txt"	Média das Distâncias Percorridas: 550.1943738719677
998882264 652459005 434370003	
Conteúdo do Arquivo "998882264.txt"	
100 200 110 205 150 180 300 200	

<p>Conteúdo do Arquivo "652459005.txt"</p> <p>3000 175 3100 176 2900 200 2800 100</p>	
<p>Conteúdo do Arquivo "434370003.txt"</p> <p>2020 1999 2018 1900 2022 1500 2020 1999</p>	

Distribuição de Pontos

Entrada – 0,5 pontos; Processamento – 3,0 pontos; Saída – 0,5 pontos.

3ª Questão (3,0 pontos)

Considere a existência de um arquivo binário chamado "meuArquivo.bin". Os primeiros 4 bytes desse arquivo armazenam um valor inteiro N que indica quantos registros o arquivo contém. Cada registro é formado por um valor numérico de ponto flutuante (4 bytes) e um campo textual com no máximo 256 caracteres (codificação UTF-8). Caracteres não utilizados são deixados em branco.

Escreva um programa que abra o arquivo indicado, leia a quantidade de registros contidos nele e ordene os N registros conforme o seguinte critério:

Seja a_1 o valor numérico e a_2 o texto que compõe um registro A qualquer e b_1 e b_2 os campos que compõe um registro B qualquer, para $A \neq B$. O registro A deverá anteceder B na ordenação se a_2 for menor que b_2 em ordem lexicográfica (ou ordem alfabética). Em caso de empate, o registro A deverá anteceder B na ordenação se a_1 for menor que b_1 .

Utilize o algoritmo *Selection Sort* na ordenação. Este algoritmo é baseado em se passar sempre o registro de menor importância para a primeira posição, depois o de segunda menor importância para a segunda posição, e assim é feito sucessivamente com os registros restantes, até os últimos dois registros.

Exemplo

Se antes de ser ordenado o arquivo "meuArquivo.bin" contém os valores

3	5.2	Gol prateado	1.4	Gol prateado	6.7	Fusca vermelho
---	-----	--------------	-----	--------------	-----	----------------

então após a ordenação o arquivo conterá os valores

3	6.7	Fusca vermelho	1.4	Gol prateado	5.2	Gol prateado
---	-----	----------------	-----	--------------	-----	--------------

Observações

Se a questão for resolvida considerando arquivos "meuArquivo.bin" textuais então a nota atribuída para a mesma será 0 (zero), mesmo que a solução esteja correta no contexto de arquivos texto.

É proibida a chamada a implementações prontas de métodos de ordenação disponíveis na API do Python 3. Será atribuída nota 0 (zero) para soluções que façam tais chamadas.

Caso seja implementado um método de ordenação diferente do *Selection Sort*, então a pontuação desse quesito será de 1,0 ponto ao invés de 1,5 pontos.

Dicas

Os tamanhos (quantidade de bytes) assumidos para formatos nativos de valores inteiros e de valores em ponto flutuante lidos ou escritos de arquivos binários podem variar de plataforma para plataforma. Ou seja, podem ocorrer problemas de compatibilidade entre programas que rodam perfeitamente em computadores que assumem determinados tamanhos para tipos primitivos, mas que não rodam corretamente em computadores que assumem outros tamanhos para o mesmo tipo. Para forçar a leitura e escrita assumindo os tamanhos padrão (standard) que são indicados na Aula 12 e ficar livre de problemas de compatibilidade, inclua o símbolo “=” na frente do formato indicado nas funções `.pack` e `.unpack` de `struct`. Por exemplo, `struct.unpack('i', bloco)` converte o bloco de bytes em um valor inteiro, mas o tamanho do bloco é dependente da plataforma (não é necessariamente de 4 bytes), enquanto que `struct.unpack('=i', bloco)` converte blocos de 4 bytes em valores inteiros, independentemente da plataforma.

Distribuição de Pontos

Leitura e escrita de arquivo binário – 1,5 pontos; Ordenação – 1,5 ponto.

Boa Avaliação!