



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação**

AD2 – 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 AD2" 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 (2,0 pontos)

Faça um programa, que contenha subprogramas, que leia da entrada padrão nomes completos, compostos de nome e sobrenome(s), de pessoas até que uma string vazia seja lida. Escreva na saída padrão todos os nomes e sobrenomes com mais de dois caracteres que ocorreram mais que uma vez. Mostre-os ordenados alfabeticamente. Neste caso implemente subprograma contendo um dos métodos de ordenação visto nas aulas gravadas.

Sugestão: Utilize listas.

Exemplo

Entrada	Saída
Carlos Drummond de Andrade Juca Kifuri Armando Carlos de Andrade e Silva Luís Carlos Kifuri Ana Silva Rolim Mariane Ribeiro	Andrade Carlos Kifuri Silva

2ª Questão (2,0 pontos)

Faça um programa, contendo subprogramas, que leia da entrada padrão o nome de dois arquivos texto a serem processados. O primeiro arquivo contém em cada linha o **nome** e o **RNA** de um coronavírus identificado, separados por um "#". O segundo arquivo representa uma população de uma cidade, contendo em cada linha o **CPF** e o **DNA** de uma pessoa, separados por um "#". Para cada coronavírus processe o arquivo da população, indivíduo por indivíduo, observando que contenha núcleos base nas mesmas posições do vírus. Os indivíduos com mais de cinquenta por cento de igualdade, letra a letra, é tida como possivelmente infectada. Escreva na saída padrão o **nome** de um coronavírus e todos os **CPF** daquelas pessoas possivelmente infectadas por aquele coronavírus. Por simplicidade, suponha que o **RNA** e **DNA** sempre possuam mesmo comprimento.

Nossa definição de RNA: É uma string de comprimento maior que zero contendo os núcleos base representados apenas por letras "A", "G", "C", "U".

Nossa definição de DNA: É uma string de comprimento maior que zero contendo os núcleos base representados apenas por letras "A", "G", "C", "T".

Exemplo

Exemplo de uma linha do primeiro arquivo (formato vírus#RNA)
Covid-19-BR-5#AGGUAAGUUUAGCACACUAAAA
Exemplo de uma linha do segundo arquivo (formato CPF#DNA)
771.822.471-86#TTGAAAGCCAGCACACTATACG

Neste exemplo, o CPF 771.822.471-86 não está infectado pelo vírus Covid-19-BR-5, pois só tem igualdade de núcleos base em 5 de 22 posições.

3ª Questão (2,0 pontos)

Faça um programa, que contenha subprogramas, que leia em uma única linha da entrada padrão o valor inteiro de um raio de distância e o nome de um arquivo texto, separados por um espaço em branco. O arquivo contém pontos do espaço bidimensional, onde cada linha possui um ponto, definido por dois números de ponto flutuante, representando coordenadas x e y. Processe todo o arquivo e escreva na saída padrão o centro geométrico, ou centroide, dado pela média de todos os x's e y's. Processe novamente o escreva todos os pontos que estejam dentro da circunferência cujo centro é o centroide e o raio é aquele lido inicialmente da entrada padrão.

Exemplo

Entrada Padrão	Arquivo pontos.txt
6 pontos.txt	9.3 20.4 9.1 4.2 4.5 5.3 1.9 2.5

Saída Padrão
Centroide: (6.2, 8.1) Listagem de Pontos dentro da Circunferência de Raio 6: 9.1 4.2 4.5 5.3

4ª Questão (2,0 pontos)

Para aumentar a arrecadação anual, o governo criou um IPPE – Imposto por Palavra Escrita. Mensalmente é divulgada uma coleção de palavras e o valor em real cobrado por seu uso em mensagens de texto. Cada mensagem escrita é então taxada conforme a quantidade de ocorrências de cada palavra na coleção, multiplicado pelo valor do importo da respectiva palavra. Mas uma medida provisória aprovada logo em seguida permite que a cobrança seja feita até, no máximo, o segundo uso de cada palavra. Para a alegria dos cidadãos, o terceiro uso em diante está liberado de cobrança.

Entrada

A entrada é dada por um arquivo binário e um arquivo texto.

O arquivo binário de nome “referencia.bin” é composto por uma sequência de registros formatos cada um por uma string que ocupa 256 bytes (uma palavra) e um número em ponto flutuante de precisão dupla (o valor do imposto sobre a respectiva palavra). O fim da sequência de registros é sinalizado pelo término do arquivo.

O arquivo texto contém a mensagem escrita que será taxada pelo novo importo. O texto nesse arquivo é composto por múltiplas linhas e as palavras podem aparecer tanto escritas com letras minúsculas quanto com letras maiúsculas. O nome do arquivo é dado pelo usuário através da entrada padrão.

As palavras poderão aparecer tanto nos registros do arquivo binário quanto no arquivo texto com letras maiúsculas quanto com letras minúsculas. Seu programa deverá ser capaz de identifica-las como iguais, independentes do caso. Ou seja, “Abacaxi”, “abacaxi” e “ABACAXI” correspondem à mesma palavra. Além disso, o arquivo texto pode incluir pontuação (ponto,

vírgula, etc.), que deverá ser ignorada pelo seu programa. Nenhuma palavra nos arquivos será acentuada.

Saída

Seu programa deverá emitir um arquivo binário de nome “imposto.bin” contendo apenas o valor do imposto a ser cobrado.

Exemplo

O exemplo que segue mostra o conteúdo dos arquivos binários na entrada e saída de forma textual apenas para facilitar a interpretação dos dados. Você deve estar ciente que os arquivos “referencia.bin” e “imposto.bin” são binários, tendo cada registro armazenado conforme a especificação.

Entrada	
Arquivo referencia.bin	Entrada Padrão
Voce 0.05 Dica 0.75 Obrigado 0.30 Tempo 1.40 Tudo 0.01 Abacaxi 0.89 EU 0.34 As 0.01 De 0.02 Novamente 1.34 Caso 0.21	uma_carta.txt
Arquivo uma_carta.txt	
Ola amigo, Quanto tempo! Tudo bem com voce? Esses dias lembrei daquela vez que eu, voce e Amanda estávamos conversando sobre as ADs do CEDERJ. Voce nos falou para fazer todas as questoes de Fundamentos de Programacao e procurar os tutores em caso de duvida. A dica que voce deu foi muito boa. Obrigado! Um abraco, nos veremos novamente no Polo.	
Saída (arquivo imposto.bin)	
4.51	

Observações

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

Dicas

Para fins de depuração de código, escreva um programa auxiliar que crie arquivos binários contendo nada mais do que valores de entrada armazenados conforme a especificação do enunciado. O programa auxiliar não deve ser entregue junto com a solução da questão. Caso seja entregue, o mesmo não será considerado na correção, independentemente de estar correto ou errado.

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.

5ª Questão (2,0 pontos)

Desenvolva e implemente algoritmos recursivos para os seguintes problemas:

- a) (1,0 ponto) Conversão e impressão de um número natural A em base binária.
- b) (0,5 pontos) Multiplicação de dois números naturais B e C, através de somas sucessivas, por exemplo, $B \times C = 6 \times 4 = 4 + 4 + 4 + 4 + 4 + 4$.
- c) (0,5 pontos) Cálculo de $1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$, onde N é um número natural diferente e zero.

No item (a), tanto a conversão quanto a impressão devem ser resolvidas com recursão. Rotinas prontas na API do Python não podem ser adotadas na conversão. Nos itens (b) e (c) o processo de repetição inerente às operações deve ser substituído por recursões de calda.

Entrada: Para cada um dos itens (a), (b) e (c) serão informados, respectivamente, um número natural A, dois números naturais B e C, e o número natural N via entrada padrão.

Saída: Seu programa deverá imprimir na saída padrão os valores calculados. Valores em ponto flutuante devem ser exibidos com quatro casas decimais.

Exemplo

Entrada	Saída
42	00101010
6 4	24
5	2.2833