



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação

AD2 1º semestre de 2017

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. Serão aceitos apenas soluções escritas na linguagem Python 3. Respostas entregues fora do formato especificado, por exemplo, em arquivos com extensão “.pdf”, “.doc” ou outras, não serão corrigidas.
 - Faça uso de boas práticas de programação, em especial, na escolha de identificadores de variáveis e subprogramas, e comentários no código.
 - As ADs deverão ser entregues pela atividade "Entrega de AD2" antes da data final de entrega estabelecida no calendário de entrega de ADs.
 - A AD é 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.
-

1ª Questão (2,0 pontos)

Considerando as operações **cons**, **car**, **cdr**, **ehLista** e **ehAtomo** vistas na Aula 8 e reescritas abaixo:

```
def car(lis):  
    return lis[0]  
  
def cdr(lis):  
    return lis[1:]  
  
def cons(x, lis):  
    return [x]+lis  
  
def ehLista(x):  
    return isinstance(x, list)  
  
def ehAtomo(x):  
    return not ehLista(x)
```

Sem utilizar a estruturas de controle de repetição (for e/ou while), sequência (suite com mais de uma linha de comando), nem fatiamento, faça os subprogramas especificados a seguir:

```
def maior(lista):
    # função que recebe como parâmetro uma lista de números e
    # retorna o maior elemento entre ele. Caso uma lista vazia
    # seja recebida retorne o valor zero.
```

Exemplos

Parâmetro de Entrada: lista
[10, 2, 5, 13, 26, 4, 2, 9, 33, 18, 6, -99, 12, 17]
[]
[-10, -13, 12.99, 100, 888, 0, 83]
Retorno da Função
33
0
888

```
def mediaPositivos(lista):
    # Função que recebe uma lista de números e retorna a média de
    # todos os números maiores que zero contidos na lista recebida.
    # Caso uma lista vazia seja recebida ou nenhum número positivo
    # seja encontrado, retornar zero.
```

Exemplos

Parâmetro de Entrada: lista
[-10, -2, -5, 13, -26, -4, 3, -9, 33, -18, -6, -99, -12, 17]
[]
[-10, -2, -5]
Retorno da Função
16.5
0
0

```
def emagrece(lista):
    # Função que recebe como parâmetro uma lista de elementos, onde
    # cada elemento pode ser um átomo ou uma lista interna. Retorna
    # uma lista simples contendo todos os átomos contidos na lista
    # recebida e todos os átomos contidos nas listas internas.
```

Exemplos

Parâmetro de Entrada: lista
[]
[[[]]]
["ana", [{"maria", "chico"}], [{"igor"}], "juca"]
[10, 2, 5, 13, [[26, [4], 2]], 9, 33, [18, 6], 99, 12, 17]
Retorno da Função
[]
[]
["ana", "maria", "chico", "igor", "juca"]
[10, 2, 5, 13, 26, 4, 2, 9, 33, 18, 6, 99, 12, 17]

Observação: Os exemplos apresentados para esta questão não preveem a leitura dos parâmetros de entrada a partir da entrada padrão. Logo, ao implementar a solução para esta questão o aluno pode assumir que a variável do programa principal que referenciar a lista de entrada é inicializada diretamente no código fonte do programa.

2ª Questão (1,5 pontos)

Errata: Havia um erro na versão original da AD2, onde o enunciado pedia para imprimir “Maior número lido=”. O correto é conforme os exemplo, onde é impresso “Maior =”.

Faça um programa que leia do usuário, via entrada padrão, o nome de um arquivo texto contendo números de ponto flutuante, onde cada linha do arquivo terá um ou mais números de ponto flutuante. Calcule e escreva o maior número contido no arquivo e a média de todos os números encontrados. Caso o arquivo esteja vazio escreva a mensagem “arquivo vazio”.

Não é permitido manter o conteúdo de todo o arquivo armazenado ao mesmo tempo na memória principal.

Entrada

A entrada é composta de uma linha contendo o nome de um arquivo texto a ser utilizado.

Saída

“Arquivo vazio” se o arquivo escolhido estiver vazio; ou duas linhas a serem escritas, onde a primeira mostra a string “Maior =” seguida de espaço em branco e o maior número encontrado, apresentado com precisão de duas casas decimais, e a segunda linha mostra a string “Média =” seguida de espaço em branco e a média calculada e apresentada com precisão de duas casas decimais.

Exemplos

Entrada	Saída
numeros.txt	Maior = 130.00 Média = 28.10

Entrada	Saída
temperaturas.txt	Maior = 45.00 Média = 38.50

Conteúdo dos Arquivos Indicados como Entrada

numeros.txt	temperaturas.txt
10 130 20 77 23 9 10 20.26 -20.26 2	45 32

3ª Questão (2,0 pontos)

Faça um programa que leia repetidamente, da entrada padrão, pares de nomes e telefones e produza um dicionário (dict), chamado de agenda. O término das entradas de nomes e

telefones deverá ocorrer quando o nome for a string “acabou”. Mostre o dicionário produzido de duas formas: ordenado pelos nomes e ordenado pelos telefones. É permitido o uso de outras estruturas de dados (list, por exemplo) para auxiliar na ordenação.

Entrada

A entrada é composta por diversas linhas. Cada linha possui um nome, formado por uma única palavra, um espaço em branco e um telefone no formato (00)0000-0000 (sem espaços em branco). É garantido que os nomes não se repetem. O término das entradas é sinalizado pela string “acabou”, que não deverá ser considerada parte da coleção.

Saída

O primeiro conjunto de saídas é composto pela impressão da agenda telefônica ordenada por nome, tendo um par nome/telefone por linha (conforme exemplo). Deve-se, então, imprimir uma linha em branco e exibir o segundo conjunto de saídas na forma da agenda telefônica ordenada por número de telefone, tendo um par nome/telefone por linha (conforme exemplo).

Exemplo

Entrada	Saída
Paulo (21)5555-1234 Ana (14)7893-9900 Claudio (11)9989-6666 Gabriela (21)1234-3452 Joao (11)5536-9786 acabou	Ana (14)7893-9900 Claudio (11)9989-6666 Gabriela (21)1234-3452 Joao (11)5536-9786 Paulo (21)5555-1234 Joao (11)5536-9786 Claudio (11)9989-6666 Ana (14)7893-9900 Gabriela (21)1234-3452 Paulo (21)5555-1234

4ª Questão (1,5 pontos)

Você foi contratado para identificar os *trend topics* do dia na famosa rede social chamada Tuiuiú-itter. A empresa que mantém o Tuiuiú-itter presta um serviço semelhante ao da rede social americana de nome parecido.

Toda manhã você recebe de seus empregadores um arquivo texto chamado “hashtags.txt” contendo as *hashtags* emitidas pelos usuários do Tuiuiú-itter nas últimas 24 horas. Sua tarefa é identificar o(s) *trend topic(s)* do dia e escrevê-lo(s) no arquivo “trendtopics.txt”.

Entrada

A entrada está contida no arquivo texto “hashtags.txt”. É fornecido um arquivo por dia e de diferentes tamanhos. Mas o nome é sempre o mesmo.

A primeira linha do arquivo contém a quantidade N de *hashtags* emitidas pelos usuários ao longo do dia. As N linhas seguintes contém as *hashtags* propriamente ditas, uma por linha.

É garantida a existência de pelo menos uma *hashtag*. Além disso, é garantido que a digitação de uma dada *hashtag* sempre segue a mesma composição de caracteres maiúsculos e minúsculos.

Saída

A saída deve ser escrita no arquivo texto “trendtopics.txt”. O arquivo deverá conter uma ou mais linhas. Tipicamente, o arquivo contém uma única linha com a *hashtag* que está na moda entre os usuários do Tuiuiú-itter. Porém, em caso de empate na quantidade de ocorrências das *hashtags* mais populares, todas as *hashtags* de maior frequência devem ser emitidas em ordem alfabética.

Exemplo

Entrada (hashtags.txt)	Saída (trendtopics.txt)
7 #BrasilNaCopa #OperacaoLavaJato #PartiuAD2 #PartiuAD2 #OperacaoLavaJato #PartiuAD2 # DietaSegundaFeira	#PartiuAD2

Entrada (hashtags.txt)	Saída (trendtopics.txt)
10 #DietaSegundaFeira #OperacaoLavaJato #BrasilNaCopa #PartiuAD2 #Carioca2017 #OperacaoLavaJato #PartiuAD2 #Carioca2017 #PartiuAD2 #Carioca2017	#Carioca2017 #PartiuAD2

5ª Questão (1,5 pontos)

Matrizes esparsas são matrizes que possuem uma grande quantidade de elementos que valem zero. Quando a matriz é grande, o armazenamento explícito desses zeros na memória torna-se custoso. Por esse motivo, dicionários (dict) são frequentemente aplicadas por bibliotecas de funções matemáticas na representação de matrizes esparsas. A estratégia é utilizar como chave os pares ordenados (i, j) que endereçam elementos diferentes de zero na matriz e associa os valores numérico em si a essas chaves. Valores iguais a zero não são armazenados, economizando, assim, muita memória.

Escreva um programa que:

- Leia duas matrizes de valores inteiros da entrada padrão e armazene-as em dicionários de maneira esparsa, isto é, desprezando os valores iguais a zero;
- Execute a multiplicação das duas matrizes, gerando uma terceira matriz também esparsa; e
- Mostre o conteúdo das duas matrizes de entrada e da matriz resultante da multiplicação na saída padrão.

Dica: O algoritmo de multiplicação de matrizes (ou produto de duas matrizes) pode ser facilmente encontrado na literatura. Mesmo assim, abaixo é apresentada a expressão matemática que calcula o valor do elemento $r_{i,j}$ (elemento na linha i e coluna j) da matriz R de tamanho $M \times N$ resultante do produto das matrizes A e B, de tamanhos $N \times T$ e $T \times M$, respectivamente, e cujos elementos são identificados por $a_{i,k}$ e $b_{k,j}$, respectivamente. Isso implica que $0 \leq i < M$, $0 \leq j < N$ e $0 \leq k < T$.

$$r_{i,j} = a_{i,0} b_{0,j} + a_{i,1} b_{1,j} + \dots + a_{i,T-1} b_{T-1,j}$$

Entrada

A primeira linha da entrada contém dois valores inteiros, N e T, que correspondem às dimensões da matriz A. As N linhas seguintes contém T valores reais em cada linha, separados por espaços em branco. O conjunto completo de valores correspondem aos elementos da matriz A.

Finalmente, é apresentada uma linha contendo dois valores inteiros, T e M, que correspondem às dimensões da matriz B. As T linhas seguintes contém M valores reais em cada, separados por espaços em branco. O conjunto completo de valores correspondem aos elementos da matriz B.

Saída

Imprima o texto “Matriz esparsa A” na primeira linha da saída padrão. Na linha seguinte, imprima o dicionário que armazena a matriz A. Imprima uma linha em branco e repita o processo para a matriz esparsa B e para a matriz esparsa R resultante da multiplicação de A e B.

Exemplo

Entrada	Saída
3 4 5 7 0 0 0 0 4 0 -6 0 0 0 4 3 0 0 0 7 0 0 0 0 6 0 2 -5	Matriz esparsa A {(0, 1): 7, (1, 2): 4, (0, 0): 5, (2, 0): -6} Matriz esparsa B {(1, 0): 7, (3, 1): 2, (3, 2): -5, (2, 2): 6} Matriz esparsa R {(1, 2): 24, (0, 0): 49}

Observação: A impressão do dicionário pode levar a ordenações diferentes dos elementos. Isso não invalida a saída. O importante é que nenhum elemento a mais ou a menos seja reportado.

6ª Questão (1,5 pontos)

Escreva um programa que recebe como entrada um arquivo binário chamado “entrada.bin” e gere dois arquivos de saída: o primeiro chamado “saida_inteiros.bin” e segundo chamado “saida_reais.bin”.

Os 4 primeiros bytes do arquivo de entrada armazenam o valor inteiro N que representa a quantidade de registros contidos no arquivo, para $N \geq 0$. O restante do arquivo de entrada corresponde aos N registros armazenados sequencialmente. Cada registro é composto por 12 bytes, sendo os 4 primeiros bytes um valor inteiro primitivo e os 8 últimos bytes um valor de ponto flutuante de precisão dupla.

A saída consiste em escrever no arquivo “saida_inteiros.bin” a quantidade inteira N , ocupando 4 bytes, seguida de todos os valores inteiros primitivos lidos de “entrada.bin”, cada um ocupando 4 bytes. De maneira análoga, a saída também consiste em escrever no arquivo “saida_reais.bin” a quantidade inteira N , ocupando 4 bytes, seguida de todos os valores de ponto flutuante lidos de “entrada.bin”, cada um ocupando 8 bytes.

Dica: Para fins de depuração de código, escreva um programa auxiliar que crie arquivos binários contendo nada mais do que valores armazenados sequencialmente conforme a especificação do enunciado, ou seja, contendo um número inteiro primitivo de 4 bytes que indica a quantidade N de registros existentes no arquivo, seguido por uma sequência de N registros de 12 bytes cada. Escreva, também, programas auxiliares que ajudem a verificar se o conteúdo dos arquivos de saída está correto. Os programas auxiliares não devem ser entregues junto com a solução da questão. Caso sejam entregues, os mesmos não serão considerados na correção, independentemente de estarem corretos ou errados.

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

Boa Avaliação!