



**Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância**

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina Fundamentos de Programação**

**APX3 2º 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 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. Respostas plagiadas não serão corrigidas. Ou seja, a questão será zerada caso seja detectado que o aluno colou, passou cola ou copiou de algum lugar. Softwares especializados serão aplicados na busca de indícios de plágio. Portanto, desenvolva sua própria solução sem a ajuda de colegas e sem ajudar colegas para não correr riscos.
-

### 1ª Questão (5,0 pontos)

Faça um programa, contendo subprogramas, que processe arquivos texto com informações parecidas com as eleições americanas. Inicialmente seu programa deve ler da entrada padrão o nome dos dois candidatos, em seguida deve ler o nome do arquivo com os nomes dos estados americanos e suas quantidades de delegados representantes. Suponha que este arquivo contém em cada linha o nome de um estado americano e a quantidade de delegados destinados a representar o partido na indicação do presidente, separados por um "#", isto é:

[nome do estado]#[quantidade de delegados]

sem colchetes.

Suponha que para cada nome de estado exista um arquivo com aquele nome, tendo em cada linha um voto com o nome do candidato. Suponha que não haja votos em branco nem nulos. Apenas dois candidatos aparecem em todos os arquivos.

Totalize a apuração em cada estado e decida qual candidato venceu naquele estado. O vencedor em cada estado fica com todos os delegados daquele estado. Por simplicidade, suponha que caso haja empate ambos candidatos recebam a quantidade de delegados (ambos venceram).

Escreva o(s) nome(s) de qual(is) o(s) estado(s) o primeiro candidato venceu, quantos delegados levou, o total de delegados e o total de votos. Em seguida, faça o mesmo para o segundo candidato.

#### Exemplo

Conteúdo dos Arquivos					
eleicoes2020	Arizona	California	Georgia	New York	Texas
Arizona#11	Trump	Biden	Trump	Biden	Trump
California#55	Biden	Biden	Biden	Trump	Trump
Georgia#16	Biden	Trump	Biden	Biden	Trump
New York#29	Trump	Biden		Biden	Biden
Texas#38	Biden	Biden		Biden	Trump
		Trump		Biden	Trump
		Biden			Trump

Entrada Padrão
Biden Trump eleicoes2020
Saída Padrão
Biden conquistou Delegado(s) no(s) Estado(s): Arizona: 11 California: 55 Georgia: 16 New York: 29 Total de Delegados para Biden: 111 Total de Votos para Biden: 16 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  Trump conquistou Delegado(s) no(s) Estado(s): Texas: 38 Total de Delegados para Trump: 38 Total de Votos para Trump: 12 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#### Distribuição de Pontos

Leitura dos arquivos – 1,5 pontos; Cômputo correto dos resultados das eleições – 2,5 pontos; Saída conforme especificado – 1,0 ponto.

## 2ª Questão (5,0 pontos)

Escreva um programa recursivo que se comporte como uma calculadora básica. Seu programa receberá como entrada uma string contendo uma expressão aritmética que incluirá apenas valores inteiros, parênteses e operações de adição (+), subtração (-), multiplicação (\*) e divisão (/). Como saída seu programa deverá emitir o resultado da expressão informada.

Lembre-se que a ordem para realizar as operações determina que multiplicações e divisões têm precedência sobre adição e subtração. Parênteses têm precedência sobre multiplicações e divisões.

### Restrições

O programa deve avaliar a expressão recursivamente. Soluções que não utilizem recursão como mecanismo de avaliação das expressões não serão consideradas.

Não é permitido importar nenhum módulo. Faça tudo utilizando Python básico.

### Dica

A avaliação de uma expressão aritmética pode ser vista como um processo recursivo, onde para avaliar cada operação antes é preciso avaliar as expressões que definem seus operandos. Por exemplo, para avaliar a expressão  $3 + 5 * 6 - 8$  podemos organizar as operações conforme a seguinte precedência dada pelos parênteses  $3 + ((5 * 6) - 8)$ . Ou seja, a chamada de função que avalia a adição do operando "3" com o operando  $5 * 6 - 8$  é chamada recursivamente para avaliar o segundo operando, que por sua vez quebra a expressão em  $5 * 6$  e  $8$  na avaliação da subtração e, em nova chamada recursiva sobre o operando  $5 * 6$  retorna o resultado da avaliação da multiplicação entre  $5$  e  $6$ .

### Descrição da Entrada

A entrada é composta por vários casos de teste, um por linha. Cada linha contém uma expressão matemática.

Sempre existirá um espaço em branco antes e depois de cada operador aritmético. Os operadores de adição, subtração, multiplicação e divisão são representados, respectivamente pelos caracteres "+", "-", "\*", e "/".

O caractere de abertura de parênteses "(" sempre será seguido de um espaço em branco. Por sua vez, o caractere de fechamento de parênteses ")" sempre será precedido de um espaço em branco.

O término da entrada é sinalizado pelo texto "0".

### Descrição da Saída

Para cada caso de teste seu programa deve imprimir uma linha contendo o resultado da expressão avaliada. Imprima os resultados com duas casas decimais.

### Exemplo

Entrada	Saída
1 + 1	2
3 + 5 * 6 - 8	25.00
( 4 + 3 ) * ( 2 - 7 )	-35.00
( ( 10 + 5 ) * 3 ) / -12	-3.75
0	

### Distribuição de Pontos

Entrada e saída – 0,5 pontos; Avaliação recursiva das expressões considerando as quatro operações básicas – 2,0; Avaliação recursiva das expressões considerando também os parênteses – 1,5.

### Questão Bônus (2,0 pontos)

*A entrega de uma solução para essa questão não é obrigatória. Ela vale pontos extras da APX3.*

Numa empresa, quando um funcionário está sobrecarregado um outro funcionário pode vir e ajudá-lo. O dono da empresa gostaria de dar um prêmio aos funcionários que mais colaboram com os colegas. Colaborar significa ajudar a um colega ou ser ajudado por um colega. Sua tarefa é fazer um programa para ajudar o dono da empresa a saber quais funcionários são mais colaborativos.

#### Atenção

Observe na distribuição de pontos da questão que soluções que utilizam ordenação como uma das etapas levam a pontuação maior. Mas, claro, o uso de ordenação precisa fazer sentido na solução. Além disso, faça e utilize seu método de ordenação. Não é permitido o uso de métodos pré-implementados em Python, tais como sort, sorted, etc.

#### Descrição da Entrada

A entrada é composta por vários casos de teste. Cada caso contém na primeira linha dois números inteiros,  $F$  ( $0 \leq F \leq 100$ ) e  $A$  ( $0 \leq A \leq 10000$ ), sendo  $F$  o número de funcionários e  $A$  o número de ajudas. Nas  $A$  linhas seguintes temos pares de números inteiros  $F1$  e  $F2$  representando que  $F1$  ajudou  $F2$  ( $F1 \neq F2 \leq F$ ). O programa termina quando  $F = A = 0$ .

#### Descrição da Saída

Para cada caso de teste seu programa deve imprimir três linhas. Na primeira linha deve ser identificado o teste com o seu respectivo número, por exemplo “Teste  $n$ ”, onde  $n$  é um valor sequencial começando em 1. Na segunda linha deve ser impresso o número correspondente ao funcionário que mais colaborou com seus colegas. Se houver empate, seu programa deve imprimir os funcionários mais colaborativos. A terceira linha deve ser deixada em branco.

#### Exemplo

Entrada	Saída
5 7 1 3 2 1 3 2 3 4 4 5 3 5 2 5 3 5 1 3 1 2 3 2 1 2 2 1 0 0	Teste 1 3  Teste 2 1 2

#### Distribuição de Pontos

Entrada e saída – 0,5 pontos; Solução do problema, mesmo sem uso de ordenação – 0,75 pontos; Uso de método de ordenação na solução – 0,75 pontos.

**Boa Avaliação!**