



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Fundamentos de Programação

APX1 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.

ERRATA

- Havia um erro no exemplo da Questão 3. O erro foi corrigido.
-

1ª Questão (3,0 pontos)

Faça um programa, contendo subprograma, que repetidamente leia da entrada padrão uma string, até que uma string vazia seja digitada. Escreva na saída padrão todas as linhas que possuam mais vogais minúsculas (“aeiou”) que dígitos (“0123456789”). Ao final também informe a quantidade de linhas lidas, excluindo a linha vazia final, e a quantidade de linhas com mais vogais minúsculas que dígitos.

Exemplos

Entrada	Saída
Abacaxi com laranja 7 reais fusão 1980 xiii gente	Abacaxi com laranja 7 reais xiii gente Total de linhas lidas: 3 Total de linhas com mais vogais que dígitos: 2

Distribuição de Pontos

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

2ª Questão (4,0 pontos)

Utilizando subprogramação, faça um programa que leia da entrada padrão uma única linha podendo conter zero ou mais números inteiros. Caso exista(m), escreva qual(is) o(s) número(s) perfeito(s). Mostre-o(s) um por linha. Caso a linha lida seja uma string vazia, escreva a mensagem: “Nenhum Número Foi Lido!!!”.

Definição

Um número inteiro é perfeito se e somente se a soma de todos os seus divisores positivos próprios (excluindo-o) é igual ao próprio número. Por exemplo, $1+2+3 = 6$; $1+2+4+7+14 = 28$.

Exemplos

Entrada
999 496 13 8 28 2 1 6 -29 37 8128 6 1024
Saída
Relação de Números Perfeitos: 496 28 6 8128 6 Fim da Relação.

Distribuição de Pontos

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

3ª Questão (3,0 pontos)

Escreva um programa que receba como entrada um número inteiro não negativo N, seguido por N linhas contendo uma palavra e um número em ponto flutuante. Após a leitura da entrada, seu programa deverá ordenar as palavras utilizando algum dos métodos de ordenação vistos em aula. A ordenação deverá considerar primeiro as palavras em ordem lexicográfica (ou ordem alfabética) crescente e, em caso de palavras iguais, considerar os valores em ordem crescente, também. Ao final, o programa deverá imprimir os pares de palavras e valores ordenados, um por linha.

Exemplos

Entrada	Saída
4 cadeira 3.2 teclado 4.5 cadeira 1.9 mesa -6.5	cadeira 1.9 cadeira 3.2 mesa -6.5 teclado 4.5

Entrada	Saída
0	

Distribuição de Pontos

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

Observação: É proibido o uso de funções de ordenação presentes na API do Python. A pontuação atribuída a essa questão será igual a zero caso tais funções sejam utilizadas.

Boa Avaliação!