Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Fundamentos de Programação

AP3 2° semestre de 2017

IMPORTANTE

- Serão aceitos apenas soluções escritas na linguagem Python 3.
- Prova sem consulta e sem uso de qualquer aparato eletrônico.
- Use caneta para preencher o seu nome e assinar nas folhas de questões e de respostas.
- Você pode usar lápis para responder as questões.
- Ao final da prova, devolva as folhas de questões e as de respostas.
- Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

1ª Questão (4,0 pontos)

Considerando a entrada de valores inteiros não negativos, ordene estes valores segundo o seguinte critério:

- Primeiro os múltiplos de 10
- Depois os demais números

Sendo que deverão ser apresentados os múltiplos de 10 em ordem crescente e depois os demais números em ordem decrescente.

Entrada

A entrada é dada via entrada padrão. A primeira linha da entrada contém o número N > 0, que indica quantos números compõe o conjunto de valores a serem ordenados. As N linhas seguintes contém os valores inteiros não negativos da coleção, um por linha.

Saída

O resultado da ordenação deve ser escrito na saída padrão, colocando-se um valor por linha.

Restrição

Não é permitido utilizar rotinas prontas de ordenação disponíveis na API do Python. Logo, você deverá implementar explicitamente um dos algoritmos de ordenação vistos em aula. Será atribuído 0 (zero) para soluções que fazem uso de rotinas prontas de ordenação.

Exemplo

Entrada Padrão	Saída Padrão
9 4 32 34 10 30 654 567 10 20	10 10 20 30 654 567 34 32 4

Distribuição de Pontos

Leitura dos dados de entrada – 20%; Ordenação – 60%; Escrita da saída – 20%.

2ª Questão (3,0 pontos)

Faça um programa, contendo subprogramas, que processe arquivos texto escolhidos pelo usuário, até que um nome de arquivo texto seja uma string “Fim”, que não deve ser processada. Para cada nome de arquivo texto lido, seu programa deve escrever a quantidade de vogais, a quantidade de consoantes e quantidade de dígitos contidos no arquivo.

Entrada

Pela entrada padrão o usuário informa N + 1 linhas, contendo em cada linha uma string. As N primeiras linhas contém os nomes de arquivos e uma linha final com a string “Fim”. Cada arquivo de nome informado é composto por texto livre.

Saída

Devem ser emitidas, na saída padrão, quatro linhas para cada arquivo, chamado aqui de xyz, da seguinte forma:

- A string “Quantidade de vogais em xyz: ” seguida da quantidade de vogais
- A string “Quantidade de consoantes em xyz: ” seguida da quantidade de consoantes
- A string “Quantidade de dígitos em xyz: ” seguida da quantidade de dígitos
- Uma linha vazia

Restrição:

Não se pode manter todo o conteúdo de um arquivo em memória principal, tais como em listas, pois deve-se assumir que os arquivos são tão grandes que tê-los completos na memória principal levaria a um erro de execução do programa.

Exemplo

Por uma questão de espaço, o conteúdo dos arquivos não é mostrado aqui.

Entrada Padrão	Saída Padrão
Ex1.txt Ex2.txt Teste.txt Fim	Quantidade de vogais em Ex1.txt : 166 Quantidade de consoantes em Ex1.txt : 294 Quantidade de dígitos em Ex1.txt : 18 Quantidade de vogais em Ex2.txt : 810 Quantidade de consoantes em Ex2.txt : 1205 Quantidade de dígitos em Ex2.txt : 84 Quantidade de vogais em Teste.txt : 207 Quantidade de consoantes em Teste.txt : 591 Quantidade de dígitos em Teste.txt : 109

Distribuição de Pontos

Fazer a leitura de todos os nomes de arquivo – 20%; Para cada arquivo, contabilizar e escrever as quatro linhas desejadas – 50%; Obedecer a restrição – 20%; Utilizar subprogramação – 10%.

3ª Questão (3,0 pontos)

Faça um programa que:

a) Peça inicialmente ao usuário um conjunto (`set`) de palavras a ser pesquisado em um arquivo, onde a escolha das palavras no conjunto termina quando o usuário escrever uma string “fim”, que não deve entrar no conjunto;
b) Escreva o conjunto na saída padrão;
c) Peça o nome do arquivo a ser consultado;
d) Processe o arquivo texto e gere um dicionário (`dict`) com contagem de ocorrência das palavras contidas no conjunto definido em (a).
e) Escreva na saída padrão, ordenadamente pelas palavras, o dicionário produzido.

Entrada

N+1 strings, uma por linha, sendo finalizada pela string “fim”. Em seguida, uma string com o nome do arquivo a ser processado.

Saída

Uma linha contendo o conjunto de palavras, seguida de zero ou mais linhas contendo a palavra do conjunto e a respectiva contagem, caso ela ocorra pelo menos uma vez. Cada linha deve ter o formato: palavra do conjunto, seguida da string “ ocorreu ”, seguida da respectiva contagem, seguida da string “ vez(es)”.

Exemplo

Entrada	
Arquivo `teste.txt`	Entrada Padrão

`o tempo perguntou pro tempo`
`quanto tempo o tempo tinha`
`o tempo respondeu pro tempo`
`que o tempo tem o mesmo tempo`
`que o tempo tinha`
	`tempo`
	`vida`
	`perguntou`
	`fim`
	`teste.txt`

Saída Padrão

`{'tempo', 'vida', 'perguntou'}`
`perguntou ocorreu 1 vez(es)`
`tempo ocorreu 9 vez(es)`

Distribuição de Pontos

Item (a) – 20%; Item (b) – 5%; Item (c) – 5%; Item (d) – 50%; Item (e) –20%.

Boa Avaliação!