

Centro de Educação Superior a Distância do
Estado do Rio de Janeiro – CEDERJ

Curso de Tecnologia em Sistemas de Computação – TSC

EAD-05.009 Fundamentos de Programação

Caderno de Exercícios

Aula 9

(Persistência de Dados – Arquivo Texto)

Professores

Dante Corbucci Filho
Leandro A. F. Fernandes

Instruções

- Utilize Python 3 e a IDE PyCharm na elaboração de soluções para os problemas propostos;
- Siga o formato apresentado na descrição da saída, caso contrário não é garantido que a saída emitida será considerada correta.

Referências Autorais

Os exercícios apresentados nesta lista foram baseados em questões sugeridas anteriormente em Avaliações a Distância (ADs) do curso de Fundamentos de Programação.

Problema A

Faça um programa que processe um arquivo texto cujo nome seja escolhido pelo usuário e escreva na tela as posições das linhas que contenham as cadeias de caracteres (*strings*) de menor e maior comprimento.

Caso o arquivo esteja vazio, escreva a seguinte mensagem na saída padrão: “O arquivo está vazio”. Caso o arquivo não exista, escreva a seguinte mensagem na saída padrão: “O arquivo não existe”.

Entrada

A entrada consiste do nome do arquivo. Esse nome só pode conter caracteres minúsculos e números, não podem haver caracteres especiais (exemplos: @, #, \$, _, >, <, *, %, etc.) nem espaços em branco. São permitidos no máximo 64 caracteres para o nome do arquivo.

Saída

A saída consiste da posição das linhas que contém a menor e a maior quantidade de caracteres, respectivamente (considere a quantidade de espaços em branco na contagem e desconsidere os caracteres “\n” e “\0”). Caso haja empate (seja para menor ou maior quantidade de caracteres) exiba a posição da última linha lida que possua aquele valor. Os valores devem ser exibidos na mesma linha, separados por um espaço em branco.

Exemplo

Entrada	Saída
tarantino.txt	5 4
burton.txt	5 6
spielberg.txt	0 arquivo está vazio
ughfiufk.txt	0 arquivo não existe

Conteúdo dos arquivos

tarantino.txt	burton.txt
Pulp Fiction\n	Edward Mãos de Tesoura\n
Kill Bill Vol. 1\n	Batman o Retorno\n
Kill Bill Vol. 2\n	Os Fantasma se Divertem\n
Bastardos Inglórios\n	Planeta dos Macacos\n
Django Livre\n	A Noiva Cadáver\n
Os Oito Odiados\n\0	O Estranho Mundo de Jack\n\0

spielberg.txt

Problema B

Faça um programa que peça ao usuário o nome de um arquivo texto e a posição de duas linhas do arquivo e, em seguida:

- (a) Mostre o arquivo na tela.
- (b) Caso seja possível, troque o conteúdo das linhas no arquivo. Caso contrário, informe ao usuário este fato com a seguinte mensagem: “Não é possível realizar a troca do conteúdo das linhas dos arquivos”.
- (c) Ao final, mostre o arquivo na tela novamente.

Caso o arquivo esteja vazio, escreva a seguinte mensagem na saída padrão: “O arquivo está vazio”. Caso o arquivo não exista, escreva a seguinte mensagem na saída padrão: “O arquivo não existe”.

Entrada

A primeira linha da entrada contém uma cadeia de caracteres T , que representa o nome do arquivo digitado pelo usuário e a segunda linha da entrada deve conter um valor inteiro $0 < X < 100$ e um valor inteiro $0 < Y < 100$, que representam as posições das linhas cujos conteúdos serão trocados, respectivamente.

Exemplo

Entrada
Teste.txt
2 4

Conteúdo dos arquivos

Teste.txt (antes do processamento)
Amarelo azul vermelho
Verde roxo laranja
Cinza preto branco
Carmim turquesa ciano

Teste.txt (depois do processamento)
Amarelo azul vermelho
Carmim turquesa ciano
Cinza preto branco
Verde roxo laranja

Problema C

Faça um programa que peça ao usuário o nome de um arquivo texto e que preencha esse arquivo com números reais positivos (o número zero e números negativos podem ser digitados, mas não devem ser gravados no arquivo), um número por linha, até que o número -1 seja. Em seguida:

1. Mostre o arquivo.
2. Leia o arquivo, e caso ele não esteja vazio, escreva na saída padrão a soma dos números, o maior número e o menor número do arquivo, respectivamente.
3. Remova todas as linhas em que ocorram o maior número e o menor número do arquivo.
4. Mostre o arquivo depois da remoção.

Entrada

A primeira linha da entrada contém uma cadeia de caracteres T , que representa o nome do arquivo digitado pelo usuário e cada uma das N linhas seguintes contém um valor do tipo ponto flutuante $-100 < X < 100$, onde X deve ter apenas 1 casa decimal. O valor -1 deve ser a última entrada.

Saída

A saída consiste de três valores do tipo ponto flutuante: o primeiro é a soma de todos os números contidos no arquivo de entrada lido, o segundo é o maior valor do arquivo e o terceiro é o menor valor contido no arquivo. Todos os valores devem ser exibidos com apenas 1 casa decimal e apresentados numa única linha, separados por um espaço em branco.

Exemplo

Entrada	Saída
arq.txt 10.5 54.3 -12.5 8.7 0.6 -4.3 58.9 0.0 -1.0 66.3 99.9 -1	299.2 99.9 0.6

Conteúdo dos arquivos

arq.txt (antes do processamento)
10.5
54.3
8.7
0.6
58.9
66.3
99.9

arq.txt (depois do processamento)
10.5
54.3
8.7
58.9
66.3

Problema D

Faça um programa que peça ao usuário o nome de dois arquivos texto contendo uma lista de nomes (cada nome em uma linha). As seguintes mensagens devem ser exibidas na leitura dos nomes dos arquivos:

“Insira o nome do primeiro arquivo:” e “Insira o nome do segundo arquivo:”

Em seguida, peça o nome de um terceiro arquivo que receberá o conteúdo dos dois arquivos anteriores alfabeticamente ordenado. A seguinte mensagem deve ser exibida:

“Insira o nome do arquivo final:”

Considere que o conteúdo dos arquivos de entrada não está ordenado inicialmente.

Exemplo

Entrada Digitada pelo Usuário
ArqA.txt
ArqB.txt
ArqFinal.txt

Conteúdo dos arquivos

ArqA.txt
Ana Caroline
Juliano Freitas
Bianca Souza
Carlos Eduardo
Jose Anisio
Carla Albuquerque
Wagner Jose
Marcos Antonio
Noelia Omena
Thiago Madureira

ArqB.txt
Rafaela Santos
Ricardo Silva
Josué da Silva
Carlos Antonio
Gabriel Limeira
Amanda Farias
Yasmin Lima
Philipe Borges

ArqFinal.txt
Amanda Farias

Ana Caroline
Bianca Souza
Carla Albuquerque
Carlos Antonio
Carlos Eduardo
Gabriel Limeira
Jose Anisio
Josue da Silva
Juliano Freitas
Marcos Antonio
Noelia Omena
Philipe Borges
Rafaela Santos
Ricardo Silva
Thiago Madureira
Wagner Jose
Yasmin Lima

Problema E

Faça um programa que crie arquivos texto, cada arquivo contendo o histórico de um único aluno, a partir de informações digitadas pelo usuário. Cada linha do arquivo texto deve conter informações sobre uma disciplina cursada pelo aluno (suponha que existam as seguintes disciplinas: matemática, português, ciências e história): o ano e o semestre em que o aluno cursou a disciplina, a carga horária da disciplina, a nota obtida e o código da disciplina.

Em seguida, exiba na tela para cada disciplina que foi cadastrada, os alunos que tiveram as maiores notas. O programa é encerrado quando o usuário digitar a palavra “fim” como nome.

Entrada

A primeira linha da entrada é composta pelo nome do aluno (esse também deverá ser o nome do arquivo). Na segunda linha deverão constar o nome da disciplina, o ano (em formato de 4 dígitos), o semestre (apenas 1 dígito), a carga horária (no máximo 4 dígitos), a nota $0.0 \leq N \leq 10.0$ e o código da disciplina (com 8 dígitos), respectivamente. Cada informação deve ser separada por um espaço em branco.

Saída

A saída consiste de várias linhas, cada uma contendo o nome da disciplina que foi cadastrada seguida do nome do aluno que obteve a maior nota e a nota que ele tirou, respectivamente. Cada informação deve ser separada por um espaço em branco.

Exemplo

Entrada	Saída
Paulo	Matemática Paulo 8.5
Matemática 2005 1 800 8.5 12345678	Português Ana 9.2
Ana	Ciências Juliana 8.9
Português 2010 2 820 9.2 87654321	História Ana 7.8
Paulo	
Português 2005 2 800 7.5 11223344	
Rafael	
Ciências 2001 1 780 6.7 36154566	
Juliana	
Ciências 2008 2 800 8.9 81273645	
Ana	
História 2010 2 800 7.8 76122345	
fim	

Conteúdo dos arquivos

Paulo.txt
Matemática 2005 1 800 8.5 12345678
Português 2005 2 800 7.5 11223344

Ana.txt									
Português	2010	2	820	9.2	87654321				
História	2010	2	800	7.8	76122345				

Rafael.txt									
Ciências	2001	1	780	6.7	36154566				

Juliana.txt									
Ciências	2008	2	800	8.9	81273645				