



Fundação CECIERJ - Vice-Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Fundamentos de Programação

AP2 1º semestre de 2017

IMPORTANTE

- Prova sem consulta e sem uso de qualquer aparato eletrônico.
 - Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas 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 (3,0 pontos)

Faça um procedimento que modifique o conteúdo de um arquivo texto por inserir uma nova linha em uma determinada posição recebida como parâmetro. Utilize o seguinte cabeçalho e retorno ("stub"):

```
def inserir(nomeArq, posLinha, novaLinha):  
    # Implemente este procedimento  
    return None
```

onde `nomeArq` é o nome do arquivo a ser modificado, `posLinha` é o número da linha onde a `novaLinha` deve ser inserida. Caso a `posLinha` seja maior que a quantidade de linhas do arquivo antes da realização desta operação, linha(s) em branco deve(m) ser inserida(s) no arquivo antes da inserção da `novaLinha`, de forma que respeite o posicionamento da nova linha.

Observação: Não é preciso escrever o programa principal. Implemente apenas o procedimento solicitado e, caso julgues necessário, implemente rotinas auxiliares.

Restrição: Não é permitido manter todo o conteúdo do arquivo na memória principal, pois o arquivo é tão grande que certamente levaria à falta de memória e término prematuro do programa.

Sugestão: Utilize um arquivo auxiliar. No entanto, não se esqueça de copiar de volta o arquivo modificado para o arquivo cujo nome é recebido como parâmetro.

2ª Questão (3,5 pontos)

Faça um programa, que contenha subprogramas, que leia do usuário o nome de um arquivo texto. Suponha que este arquivo contenha em cada linha uma ou mais palavras separadas por espaço em branco, e produza um dicionário (`dict`) de todas as palavras que comecem

com vogais minúsculas e suas respectivas contagens de ocorrências no arquivo. Ao final escreva na saída padrão o dicionário produzido.

Observação: Implemente o programa principal e subprogramas nesta questão.

Restrições:

- (1) O arquivo deve ser lido apenas uma vez.
- (2) Não é permitido manter todo o conteúdo do arquivo na memória principal, pois o arquivo é tão grande que certamente levará à falta de memória e término prematuro do programa. Portanto, não utilize lista para manter todo o conteúdo do arquivo na memória principal. Felizmente, há memória suficiente para se manter um dicionário de palavras com a contagem de ocorrências. Códigos que não cumprem essa restrição não serão considerados corretos.

Dica: Os caracteres de uma string podem ser acessados a partir de colchetes. Por exemplo, se a string "abacaxi" está associada à variável `s`, então `s[0]` retorna "a".

3ª Questão (3,5 pontos)

Considere a existência de um arquivo binário chamado "entrada.bin". Os primeiros 4 bytes desse arquivo armazenam um valor inteiro N que indica quantos valores numéricos de ponto flutuante de precisão dupla (cada valor composto por 8 bytes) existem no restante do arquivo.

Escreva um programa que:

- (1) Abra o arquivo binário "entrada.bin" e leia os N valores para dentro de uma lista (`list`). Nesta questão você pode assumir que o arquivo não é muito grande, de modo que todo o seu conteúdo cabe na memória ao mesmo tempo.
- (2) Ordene os valores na lista de forma crescente utilizando o algoritmo *Selection Sort*. Este algoritmo é baseado em se passar sempre o valor de menor importância para a primeira posição, depois o de segunda menor importância para a segunda posição, e assim é feito sucessivamente com os valores restantes.
- (3) Escreva os valores ordenados em um arquivo binário de saída chamado "saida.bin". Esse arquivo segue o mesmo formato interno que o arquivo de entrada. Ou seja, os primeiros 4 bytes desse arquivo armazenam um valor inteiro N que indica quantos valores numéricos de ponto flutuante de precisão dupla (cada valor composto por 8 bytes) existem no restante do arquivo.

Exemplo

Se antes de ser ordenado o arquivo "entrada.bin" contém os valores

6	5.5	9.0	5.4	7.2	1.0	3.5
---	-----	-----	-----	-----	-----	-----

então após a ordenação o arquivo conterá os valores

6	1.0	3.5	5.4	5.5	7.2	9.0
---	-----	-----	-----	-----	-----	-----

Observação: Os arquivos são binários. A entrada e saída de programas que consideram arquivos texto não serão considerados na correção.

Restrição: Não é permitido o uso de rotinas de ordenação disponíveis na API Python, tal como o método `sort()` de `list`. Você deve implementar sua própria rotina de ordenação conforme o algoritmo de *Selection Sort*, cuja descrição é apresentada em vídeo-aula e visão geral, suficiente para implementação, é apresentada neste enunciado. A implementações de *Bubble Sort* ou de *Quick Sort* também será aceita na correção, sem descontos, caso correta.

Boa Avaliação!