

ACH2023 ALGORITMOS E ESTRUTURAS DE DADOS I

Semestre 2010-1 - Exercício prático 1 – Codificação de strings – t.94

Descrição do EP: A partir do projeto exemplo disponibilizado no COL (*projetoPalavras.dev*), editar o módulo *trabalho.cpp* implementando as seguintes operações para manipulação de frases de até 150 caracteres:

1. O projeto exemplo fornecido contém uma função *main.cpp* que pode ser usada para execução e teste do seu programa (já contendo os tipos de dados manipulados e outras especificações globais) e o módulo *trabalho.cpp* onde deve ser realizado o trabalho propriamente dito.
2. O objetivo do trabalho é implementar de forma correta e completa a função **codificar** utilizando apenas listas ligadas como estrutura de armazenamento.
3. A função recebe como entrada uma lista ligada de caracteres representando uma frase, e retorna a mesma frase codificada da seguinte forma¹: Em uma 1ª. etapa, invertemos as seqüências de não vogais, incluindo espaços e pontuação. Em uma 2ª. Etapa, invertemos a mensagem resultante. Exemplo:

Dada a mensagem:	ESTRUTURAS DE DADOS É MUITO LEGAL.
Após a 1ª etapa, teremos:	ERTSUTURAD SED ADO SÉM UITOL EGAL.
E depois da 2ª etapa teremos:	LAGE LOTIU MÉS ODA DES DARUTUSTRE

4. A assinatura da função é a seguinte:
NO* codificar(NO *frase)
5. A frase a ser fornecida como entrada será sempre válida e não-vazia, contendo apenas letras sem acentuação, espaços e pontuação.
6. Não use nenhum vetor na sua implementação. Se necessitar de estruturas auxiliares, use sempre listas ligadas de implementação dinâmica.
7. Não exiba nenhuma mensagem na tela, nem solicite que o usuário pressione nenhuma tecla etc. Apenas implemente a função solicitada.
8. A função implementada deve estar inteiramente contida no módulo *trabalho.cpp*. Note no entanto que para testar a sua implementação e garantir sua correção você provavelmente terá de criar várias outras funções auxiliares (e.g., entrada de dados, exibição etc.) que não fazem parte do trabalho e que não serão avaliadas.
9. Observe que tudo que for necessário para executar a função solicitada deve obrigatoriamente estar em *trabalho.cpp*, ou seu EP estará incompleto.
10. Além da função acima, solicita-se que o aluno complete as funções *aluno1*, *aluno2*, *nrousp1* e *nrousp2* e *turma* também presentes no módulo *trabalho.cpp* para fins de identificação dos autores. Para trabalhos individuais, deixe as informações do segundo autor em branco.
11. **IMPORTANTE:** o seu programa será corrigido de forma *automática*, e por isso você não pode alterar as assinaturas da função solicitada, nem os tipos de dados ou especificações em *palavras.h*.

¹ Adaptado de material didático desenvolvido pela Profa. Maria das Graças Volpe Nunes (USP//ICMC)

12. Este trabalho pode ser desenvolvido *individualmente ou em duplas*, desde que estas sejam *confirmadas em aula até 3 semanas antes da entrega*. Em ambos os casos, não tente emprestar sua implementação para outros colegas/duplas, nem copiar deles, pois isso invalida o trabalho de todos os envolvidos.
13. O programa deve ser compilável no Dev-C++ versão 4.9.9.2. sob Windows XP ou Vista.

O que entregar:

Apenas o arquivo *trabalho.cpp* extraído do projeto implementado, observando que ele deve conter todas as rotinas necessárias para a execução da função solicitada. A entrega será realizada via sistema COL até a data e hora de início da P2. EPs entregues após este horário serão desconsiderados.

Critérios de avaliação:

A função será testada com uma série de 10 chamadas consecutivas; cada codificação correta (i.e., resultando em um string que possa ser decodificado em uma frase idêntica à original) vale 1,0 ponto.

Este EP deve ser desenvolvido obrigatoriamente por *todos* os alunos de AED1. Sua nota faz parte da 2ª. avaliação da disciplina, que é calculada como o *mínimo* entre P2 e EP. A nota do EP não é passível de substituição e, caso este não seja entregue, implica nota zero na 2ª. avaliação independentemente do resultado na P2 ou SUB.