



DISCIPLINA: Laboratório de Programação I

PROFESSOR: Julio Cesar Duarte

VISTO: _____

ANO / CURSO: 3º / Computação

DATA DA REALIZAÇÃO: 09 / 06 / 2022

TIPO DE PROVA: VE2 - Prática

DURAÇÃO: 180 min

HORA: 07:30

CONSULTA:

LIVRE

X

RESTRITA

PROIBIDA

1. (5,0) Implemente uma classe em C++ para o Jogo da Forca. Nesse jogo, você possui uma quantidade de chances limitadas para acertar as letras de uma palavra. O jogo termina com vitória se a palavra for completa antes das chances terminarem e com derrota se as chances acabarem antes. A sua implementação deve conter

(a) (1,0) classe, construtor(es), destrutor(es) e atributo(s) necessários. No construtor, o único parâmetro passado é uma lista de palavras a serem utilizadas no jogo. Obrigatoriamente utilize um *container* da STL para armazenar as palavras. O número de chances máximo deve ser um atributo compartilhado por todas os objetos e que não pode ser alterado.

(b) (1,0) um método **iniciar** que zera as chances utilizadas, e escolhe uma palavra aleatoriamente da lista atual. Após isso, a palavra não pode ser mais escolhida;

(c) (1,0) um método **mostrar** que apresenta a palavra construída até o momento e o número de chances restantes;

(d) (1,0) um método **jogar** que recebe a entrada do usuário (uma letra possível). Uma jogada é válida apenas para letras não repetidas e uma chance deve ser consumida, caso a letra não esteja na palavra.

(e) (1,0) um método **testeFimdeJogo** que retorna um booleano indicando se o jogo acabou ou não. Antes de retornar verdadeiro, ele deve imprimir na tela uma mensagem de vitória ou derrota;

2. (5,0) Implemente um modelo chamado `vetorOrdenado` que armazena valores comparáveis de forma ordenada. Sua implementação deve conter para os modelos:

- a) (1,0) classes, construtor(es), destrutor(es) e atributo(s) necessários. Obrigatoriamente utilize um *container* da STL para armazenar os valores. O `vetorOrdenado` possui uma capacidade especificada durante o construtor;
- b) (1,0) leitura dos valores do vetor de um arquivo, utilizando o operador `<<` seguido pelo nome do arquivo. Esse arquivo contém os valores separados por vírgulas (todos na mesma linha). Caso o arquivo tenha mais valores que a capacidade do vetor, o excedente deve ser não considerado;
- c) (1,0) busca da posição de um valor no `vetorOrdenado`, utilizando o operador `+` seguido do valor. Utilize obrigatoriamente um algoritmo da STL para a busca;
- d) (1,0) contagem da quantidade de elementos do `vetorOrdenado` menores que um certo valor, utilizando o operador `-` seguido do valor. Utilize obrigatoriamente um algoritmo da STL para a contagem;
- e) (1,0) integração do `vetorOrdenado` com o `cout` (operador `<<`) de forma a imprimir o seu conteúdo na tela.

Observações gerais para a realização da prova

- 1) As questões devem ser enviadas pelo *classroom*, em sua atividade correspondente, no seguinte formato: `ve_codigo_do_aluno_Q1.cpp` e `ve_codigo_do_aluno_Q2.cpp`
- 2) As questões serão avaliadas quanto à simplicidade das soluções, adequação ao enunciado e melhor utilização das técnicas de orientação a objeto ensinadas. Não são permitidos a utilização de variáveis globais, nem implementação de funcionalidades externamente à classe (a não ser quando estritamente necessário).
- 3) Todos os atributos das classes devem ser não-públicos e devem ser providos métodos para a sua manipulação (leitura e escrita, quando necessário)
- 4) Preste atenção para evitar problemas no seu programa, mesmo que eles não causem impacto no resultado gerado, especialmente àqueles relacionados à alocação dinâmica. **Dica:** construtores por cópia, operadores de atribuição, etc..
- 5) Junto à classe, deve ser fornecido uma função `main` que teste todos os pontos sendo avaliados do programa
- 6) Questões enviadas que não compilem ou que imediatamente apresentem erros de execução serão penalizadas severamente
- 7) A prova é individual e a consulta é exclusiva ao material relacionado à linguagem C++