Lista 4 - Vector/Classes/Algorithm   
Técnicas de Programação 2019.2

Programação Estruturada 2019.2

linha curta

# Resumo

A essa altura do curso nós já vimos muitos conceitos novos e você já deve estar mais familiarizado com a ideia de orientação a objetos e a sintaxe do C++, é hora então de exercitar pensando no projeto final.

Nesta lista vamos supor que temos um “sensor” genérico que está gerando uma série de dados e que precisamos organizar esses valores e obter informações sobre eles. Vamos começar gerando um conjunto de dados aleatórios, para simular o nosso sensor, e depois criar uma Classe que nos ajude a organizar esses dados com auxílio da Classe Vector.

A lista foi pensada para ter questões que vão progressivamente se tornando difíceis, de questões bastante simples até muito desafiadoras. As questões são dependentes entre si, o que significa que você deverá resolver a lista na ordem em que as questões são apresentadas.

*“C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off.”*

--Bjarne Stroustrup

Have fun!

Form de entrega: <https://forms.gle/Ydd84wjsx8AMUiA49>

O formulário ficará aberto para entrega até o dia 24/10/2019.

# Questões

Q1. Faça um programa que seja capaz de receber um número indefinido de valores inteiros, armazenar em vetor e imprimir na tela os valores lidos.

Q2. Faça um programa que gere 500 números aleatórios entre 0 e 999 e salve os valores em um arquivo .txt.

Dica: *Pesquise sobre o uso das funções rand, srand. Faça o programa de uma forma que seja fácil de alterar a quantidade de valores gerados para facilitar o processo de debug do programa.*

Q3. Modifique o programa da questão 1 para que ele passe a ler os valores inteiros do arquivo criado na questão 2. Lembre-se de armazenar os valores em um vetor e imprimir na tela somente depois de feita a leitura de todos os números.

Q4. Modifique o programa da questão 3 para que ele mostre na tela qual o maior e qual o menor número lidos do arquivo.

Dica: *Você pode usar a função sort() ou pesquisar mais sobre a biblioteca <algorithm>.*

Q5. Adicione ao programa da questão 4 a capacidade de contar quantas vezes cada valor aparece no arquivo lido, ou seja, a frequência com qual esse valor ocorre no arquivo.

Q6. Crie um classe chamada **DadosSensor** que contenha

* + atributos privados:
    1. dois inteiros: **valor** e **frequencia**;
  + como métodos:
    1. um construtor que inicializa os atributos privados;
    2. **int getValor()**
    3. **int getFrequencia()**
    4. **void printDados()**
  + Faça um programa para mostrar as funcionalidades da classe **DadosSensor**.

Q7. Modifique a questão 5 para fazer uso de um vetor da classe **DadosSensor.**

Q8. Altere o programa da questão 7 para que ele mostre na tela os valores ordenados de forma **decrescente** de acordo com a contagem da frequência de cada número.

Q9. Modifique o programa da questão 8 de forma que seja possível salvar em um arquivo **.txt** duas colunas que indicam o valor e a frequência que ele aparece no arquivo inicial. Os valores devem ser salvos de forma decrescente de acordo com a frequência e os valores com frequência 0 não devem ser salvos.

* + *Dica: Faça uso de iteradores para acessar os métodos necessários dentro do loop.*
  + Exemplo de como o arquivo deve ser salvo:

**Valor Frequência**

**996 4**

**567 4**

**368 4**

**340 4**

**500 3**

**754 3**

**743 3**

**729 3**

Q10. Faça um programa de carrega os dados do arquivo salvo na questão 9 e mostre na tela apenas os 10 primeiros pares de valores.

Q11. Modifique o programa da questão 10 para que seja possível pesquisar um número dentro do arquivo e indicar quantas vezes ele apareceu no arquivo original. Caso o valor não seja encontrado o programa deve avisar ao usuário que aquele valor não existe.