

TRABALHO PRÁTICO 2

Depois de aprender a usar **vetores dinâmicos** e **laços de repetição** no curso de Programação de Computadores, Joãozinho começou a procurar novos desafios e decidiu testar suas habilidades criando uma competição de programação com seus amigos.

A competição consiste em classificar e resolver um determinado número de questões de programação. O tempo de solução é um ponto importante para a competição e cada participante deverá preencher uma ficha durante o concurso indicando para cada questão:

- Dificuldade da questão: valor entre 1 (muito fácil) e 5 (muito difícil)
- Horário de início da resolução (no formato horas e minutos)
- Horário de fim da resolução (no formato horas e minutos)

Ao final do concurso, Joãozinho quer mostrar aos participantes algumas estatísticas. Ajude Joãozinho escrevendo um programa que colete as informações das fichas e apresente um sumário do concurso, como mostrado abaixo.

```
*****
Concurso de Programação

Qual o número de participantes? 2
Qual o número de questões? 2

*****
-----
Participante: João
-----
Questão A
    Dificuldade: 1
    Início: 9:30
    Fim: 9:45
Questão B
    Dificuldade: 4
    Início: 9:50
    Fim: 10:20
-----

-----
Participante: Pedro
-----
Questão A
    Dificuldade: 2
    Início: 10:00
    Fim: 10:12
Questão B
    Dificuldade: 5
    Início: 10:30
    Fim: 11:05
-----
```


Resumo por Participante

João:
A (1) 09:30 às 09:45 (15 min)
B (4) 09:50 às 10:20 (30 min)
Pedro:
A (2) 10:00 às 10:12 (12 min)
B (5) 10:30 às 11:05 (35 min)

Resumo por Questão

Questão A:
João (1) 09:30 às 09:45 (15 min)
Pedro (2) 10:00 às 10:12 (12 min)
Questão B:
João (4) 09:50 às 10:20 (30 min)
Pedro (5) 10:30 às 11:05 (35 min)

Estatísticas

Questão A: Dificuldade (1.50) - Tempo (13.50 minutos)
Questão B: Dificuldade (4.50) - Tempo (32.50 minutos)
Concurso: Dificuldade (3.00) - Tempo (23.00 minutos)

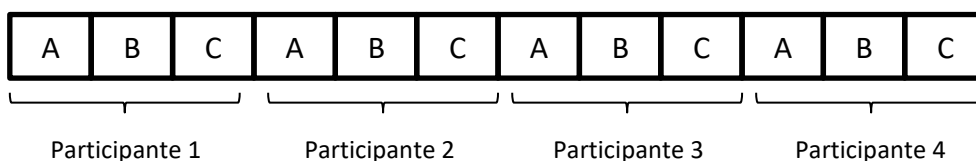
INSTRUÇÕES

O programa deve pedir o número de participantes e o número de questões e usar essas informações para criar um **vetor dinâmico de questões**, que conterà as informações coletadas a partir da ficha do concurso.

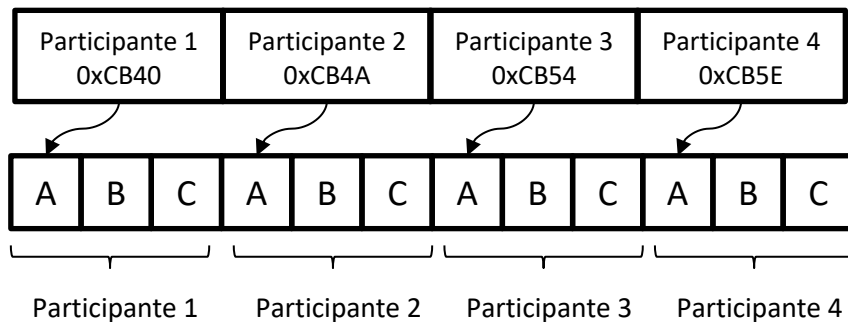
Armazene os dados do programa com as seguintes estruturas:

- Registro para representar um horário (hora e minuto)
- Registro para representar as questões (grau de dificuldade, início e fim)
- Registro para representar uma média (grau de dificuldade e tempo)
- Registro para representar participantes (nome e ponteiro para questões)
- Vetor dinâmico de questões
- Vetor dinâmico de participantes

O vetor de questões deverá guardar todas as questões do concurso. Supondo que houvessem 4 participantes e 3 questões (A, B e C), o vetor teria a seguinte organização dos dados:



Cada elemento do vetor de participantes deve conter o nome do participante e o endereço onde iniciam suas informações no vetor de questões.



Para resolver o problema, construa e use pelo menos as seguintes funções:

- Uma função que receba um vetor de questões e uma quantidade de questões a exibir. A função deve mostrar uma letra para cada questão (iniciando em A), a sua dificuldade, horário de início, horário de fim e o tempo calculado da solução em minutos. As questões exibidas devem estar em posições consecutivas do vetor. Use a função para gerar o resumo por participante. Ela deve ser chamada uma vez para cada participante. O nome do participante não deve ser mostrado pela função.
- Uma função que receba o vetor dinâmico de questões, o tamanho do vetor, o índice inicial do elemento que será contabilizado no resultado e o tamanho do passo para obter o índice do próximo elemento. A função deve retornar uma média (um registro tipo média contendo a média de dificuldade e de tempo para os elementos analisados). A função deve ser chamada uma vez para cada questão e uma última vez para obter as estatísticas totais do concurso.
- Para gerar os traços separando as seções do programa, construa e use uma função que receba um caractere e um valor inteiro representando o tamanho da linha a ser exibida.
- Faça com que horários (registro horário) possam ser lidos com cin, exibidos com cout e subtraídos com o operador de subtração, operador - (). O resultado da subtração deve ser um valor representando a diferença em minutos. Para simplificar, considere que os horários estão sempre dentro de um mesmo dia.

Separe o programa em três arquivos:

- **estatisticas.cpp**: programa principal com a função main()
- **concurso.h**: arquivo de cabeçalho contendo as definições dos registros e os protótipos das funções do programa
- **concurso.cpp**: contendo a implementação das funções definidas em concurso.h

Exigências de organização do trabalho:

- 1) Não use variáveis globais
- 2) Use comentários, indentação e deixe o código organizado
- 3) Os vetores dinâmicos devem ser criados dentro da função main()
- 4) O arquivo estatisticas.cpp deve conter apenas uma função, a função main()

ENTREGA DO TRABALHO

Grupos: Trabalho individual

Data da entrega: 05/02/2019 (até a meia noite)

Valor do Trabalho: 3,0 pontos (na 2a Unidade)

Forma de entrega: enviar apenas os arquivos fonte (.cpp) e os arquivos de inclusão (.h) compactados no formato **zip** através da tarefa correspondente no SIGAA.

O não cumprimento das orientações resultará em **penalidades:**

- Programa não executa no Visual Studio 2017 (3,0 pontos)
- Programa contém partes de outros trabalhos (3,0 pontos)
- Atraso na entrega (1,5 pontos por dia de atraso)
- Arquivo compactado em outro formato que não zip (0,5 ponto)
- Envio de outros arquivos que não sejam os .cpp e .h (0,5 ponto)
- Programa sem comentários e/ou desorganizado (0,5 ponto)