ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

PCS 3111 - Laboratório de Programação Orientada a Objetos para Engenharia Elétrica

2021

Aula 01 – Introdução

Cuidados

1. Os tipos, os nomes, e os parâmetros das funções **devem seguir o especificado** em cada exercício para fins de correção automática**.**
2. A função `main` **não deve ser submetida**. Caso contrário, a correção automática retornará um *Compilation Error.*
3. Use o código fornecido no Discord.

Exercício 1
Implemente a função:

```
int calcularNota(int numeroDeReproducoes, int quantidadeDePositivos, int
                    quantidadeDeUsuariosQueOuviram)
```

que retorna a nota (avaliação de 0 a 5) de uma determinada música. A função recebe três parâmetros: (i) `int numeroDeReproducoes` que indica o número de reproduções daquela música, (ii) `int quantidadeDePositivos` que indica o número de avaliações positivas, e (iii) `int quantidadeDeUsuariosQueOuviram` que indica a quantidade de usuários diferentes que ouviram a música.

A nota em função dos parâmetros dados é a apresentada abaixo:

Nota 5: 1.000.000 reproduções ou mais;
Nota 4: 100.000 reproduções ou mais, ou 50% de positivos/usuários ou mais;
Nota 3: 10.000 reproduções ou mais, ou 35% de positivos/usuário ou mais;
Nota 2: 1.000 reproduções ou mais, ou 20% de positivos/usuário ou mais;
Nota 1: 100 reproduções ou mais, ou 15% de positivos/usuário ou mais;
Nota 0: menos de 100 reproduções, e 15% de positivos/usuário ou menos;

A nota deve ser a maior possível, ou seja, uma música que teve 10.000 reproduções e 60% de positivos/usuários deve ter nota 4.

O valor de *positivo por usuários que ouviram* é dado pela razão da quantidade de avaliações positivas pelo **número** de usuários diferentes que escutaram a música. Como essa razão é feita por variáveis do tipo inteiro, é necessário utilizar um `double` para obter as casas decimais, como no exemplo a seguir:

```
int a = 5, b = 10;
double x = ((double) a) / b;
```

Exemplo: `calcularNota(1500000, 450000, 800000)` deve retornar `5`.
`calcularNota(120000, 10000, 70000)` deve retornar `4`.

Exercício 2
Deseja-se calcular o tempo total de músicas que possuem uma determinada nota em uma *playlist* (uma lista de músicas). Para isso implemente a função:

```
int calcularTempoTotal(int nota, int notas[], int duracoes[], int quantidade)
```

O parâmetro `nota` indica a nota pela qual deve-se basear a busca, a `quantidade` representa o número de músicas que fazem parte da *playlist*. São passados ainda dois vetores: um que contém as notas, e outro a duração de cada música pertencente à *playlist*.

Considere que a playlist sempre terá músicas (ou seja, quantidade > 0). Por exemplo, suponha que se deseja o tempo total das músicas com nota 4 em uma playlist que possua 3 músicas, onde:

- 1° música: nota 4 e duração 5 minutos.
- 2° música: nota 5 e duração 3 minutos.
- 3° música: nota 4 e duração 4 minutos.

Nesse caso, `nota` é 4, os vetores `notas` e `duracoes` são {4, 5, 4} e {5, 3, 4}, respectivamente, a `quantidade` vale 3, e o tempo total resulta em 9 minutos. Caso não haja músicas com a nota na *playlist*, o método deve retornar 0.

Exercício 3
Agora vamos implementar a seguinte função:

```
bool temRepetido(string artistas[], int quantidade)
```

que recebe como parâmetros um vetor com os nomes dos artistas que possuem músicas em uma playlist e o número de elementos deste vetor. A função retorna `true` caso exista pelo menos um nome repetido no vetor e `false` caso não existam nomes repetidos.

Exemplo:

- {“**Djonga**”, “Alok”, “Ferrugem”, “Seu Jorge”, “**Djonga**”} e quantidade 5 deve retornar `true`.
- {“Ferrugem”, “**Djonga**”, “**Alok**”, “**Djonga**”, “**Alok**”, “Seu Jorge”, “**Alok**”} e quantidade 7 deve retornar `true`.
- {“Djonga”, “Alok”, “Ferrugem”, “Seu Jorge”} e quantidade 4 deve retornar `false`.

Testes do Judge
Exercício 1

- calcularNota nota 5
- calcularNota nota 4
- calcularNota nota 3
- calcularNota nota 2
- calcularNota nota 1
- calcularNota nota 0

Exercício 2

- calcularTempoTotal com apenas uma música com a nota desejada

- calcularTempoTotal com várias músicas com a nota desejada
- calcularTempoTotal com nenhuma musicas com a nota desejada

Exercício 3

- temRepetido para vetor com nenhum nome igual:
- temRepetido para vetor com um nome igual cujas posições são consecutivas:
- temRepetido para vetor com um nome igual cujas posições não são consecutivas:
- temRepetido para vetor com mais de 2 nomes iguais: