QUESTÃO 02 — Arquivos em C/C++ (Modo Texto e Modo Binário)

Considere o sistema de manutenção de contas bancárias implementado na questão anterior, no qual os dados dos clientes são armazenados em um arquivo utilizando acesso direto. Cada registro é representado pela seguinte estrutura:

```
struct cliente {
    int numero;
    char nome[30];
    double saldo;
};
```

O programa utiliza as funções `read()` e `write()` para manipular os registros no arquivo, posicionando o cursor com as funções `seekg()` e `seekp()`.

Responda:

1. Explique a diferença entre abrir um arquivo em modo texto e em modo binário em C/C++.

> Modo texto permite conversões automáticas de caracteres (ex.: \n → \r\n no Windows). Modo binário lê e grava os dados byte a byte, sem qualquer alteração pelo sistema.

2. Justifique por que, nesse sistema, o arquivo deve obrigatoriamente ser aberto em modo binário (`ios::binary`).

> Porque o sistema grava e lê estruturas diretamente no arquivo usando acesso direto. O modo binário garante que o tamanho e a posição dos registros não sejam alterados.

3. O que pode acontecer se o arquivo for aberto apenas com `ios::in | ios::out`, sem o uso de `ios::binary`, especialmente em sistemas Windows?

> Podem ocorrer conversões automáticas de bytes, alterando o tamanho real dos dados. Isso causa desalinhamento de registros, leituras incorretas e possível corrupção do arquivo.

4. Explique a diferença entre as funções `seekg()` e `seekp()` e descreva um erro comum cometido ao utilizá-las incorretamente.

> seekg() posiciona o cursor de leitura e seekp() o cursor de escrita. Erro comum: usar seekg() antes de write() ou seekp() antes de read().

QUESTÃO 03 — Organização de Programas em C/C++ com Arquivos `.h` e `.cpp`

Em C/C++, é comum organizar programas grandes utilizando **arquivos de cabeçalho** (`.h`) e **arquivos de implementação** (`.cpp`). Considere o seguinte cenário:

- O arquivo `funcoes.h` contém declarações de funções.
- O arquivo `funcoes.cpp` contém as implementações dessas funções.
- O arquivo `main.cpp` utiliza essas funções através de `#include "funcoes.h"`.

Durante a compilação no Code::Blocks, o seguinte erro é apresentado:

undefined reference to 'openFile'

Com base nesse contexto, responda:

1. Explique o que significa o erro **undefined reference**.

> Indica que o compilador encontrou a declaração da função,
> mas o vinculador não encontrou sua implementação durante a linkdição.

2. Cite duas causas comuns para esse erro ao trabalhar com múltiplos arquivos em C++.

> O arquivo `.cpp` que contém a implementação não foi incluído no projeto,
> ou a assinatura da função na implementação é diferente da declaração no `.h`.

3. Explique a função das diretivas `#ifndef`, `#define` e `#endif` em arquivos de cabeçalho.

> Elas evitam varias inclusões do mesmo arquivo de cabeçalho,
> impedindo erros de redefinição durante a compilação.

4. Justifique por que as implementações das funções não devem ser colocadas diretamente no arquivo `.h`.

> Isso pode causar varias definições da mesma função em projetos grandes,
> além de aumentar o tempo de compilação e dificultar a manutenção do código.