

Trabalho de Laboratório: Monitoramento de Dados em uma Smart City

Disciplina: Introdução à Programação – Laboratório

Tema: Revisão de Programação – Estruturas, Loops e Matrizes

Introdução

Imagine uma **cidade inteligente (Smart City)** onde sensores espalhados monitoram diferentes bairros para coletar dados sobre o clima, o nível de poluição e o trânsito. Esses dados precisam ser analisados para ajudar os gestores da cidade a tomar decisões.

Sua tarefa será escrever um **programa em C++** que **armazene, processe e exiba** os dados simulados desses sensores.

Descrição do Problema

O programa deverá armazenar as seguintes informações de **5 bairros da cidade ao longo de 7 dias (segunda a domingo)**:

1. **Temperatura diária (°C):** Deve variar entre 5 e 40 graus.
2. **Nível de poluição (ppm de CO₂):** Deve variar entre 200 e 600.
3. **Tráfego (em %):** Deve variar entre 0 (trânsito livre) e 100 (congestionamento total).

Estas informações serão armazenadas em três **matrizes** de 5x7 (5 bairros x 7 dias). Os dados serão gerados automaticamente usando números aleatórios.

Tarefas Passo a Passo

1. Criar as Matrizes

No início do programa, você deve criar:

- Uma matriz para armazenar os dados de temperatura;
- Uma matriz para o nível de poluição;

- Uma matriz para o tráfego.

Todas as matrizes terão o tamanho **5 x 7**. Use funções da biblioteca `<cstdlib>` para preencher estas matrizes com valores aleatórios.

Exemplo de Geração de Dados Aleatórios: Temperaturas entre 5 e 40.

```
temperatura[i][j] = rand() % 36 + 5; // Entre 5 e 40
```

2. Exibir os Dados Gerados

Exiba os dados da matriz de forma **tabular**, para que fiquem organizados. Um exemplo para a matriz de temperatura seria:

```
Temperaturas (°C):
Bairro  Dia 1  Dia 2  Dia 3  Dia 4  Dia 5  Dia 6  Dia 7
1        20    22    25    30    35    34    29
2        18    20    21    22    25    27    26
...      ...    ...    ...    ...    ...    ...    ...
```

3. Processar os Dados

Agora que as matrizes estão preenchidas, processe os dados e calcule as seguintes informações:

A. Média Semanal de Temperatura (em cada bairro):

- Para cada bairro, calcule e exiba a temperatura média ao longo dos 7 dias.

B. Bairro Mais Quente da Semana:

- Compare as médias das temperaturas entre os bairros para determinar qual teve a maior média.

C. Dia com Maior Poluição:

- Compare os valores da matriz de poluição para encontrar o **maior nível de CO₂** entre todos os bairros em todos os dias. Indique qual foi o dia e o bairro.

D. Média Semanal de Tráfego (de cada bairro):

- Para cada bairro, calcule e exiba a **média de tráfego** ao longo dos 7 dias.

E. Bairro com o Maior Tráfego Total:

- Some os valores de tráfego de cada bairro e determine qual teve o maior valor total.

4. Relatórios

Seu programa deve exibir:

1. As matrizes com os dados gerados;
2. Os resultados dos cálculos acima (de forma explicativa e clara, como em relatórios).

Exemplo de relatório:

```
=== Relatório de Análise da Semana ===
1. Média semanal de temperatura:
  - Bairro 1: 21.5 graus
  - Bairro 2: 19.8 graus
  ...
2. Bairro mais quente da semana: Bairro 3, com 26.4 graus.
3. Dia com maior nível de poluição: quarta-feira no Bairro 2, com 580
ppm CO2.
4. Média semanal de tráfego:
  - Bairro 1: 60.4%
  - Bairro 2: 55.8%
  ...
5. Bairro com maior tráfego registrado: Bairro 1, com 450% de tráfego
acumulado.
```

Estrutura do Código

Segue um esquema básico para ajudá-lo na organização do código:

1. **Declarações de bibliotecas e variáveis:**

Inclua as bibliotecas necessárias (`<iostream>`, `<cstdlib>`, `<ctime>`).

2. **Declaração das matrizes:**

Três matrizes de tamanho 5x7.

3. **Preenchimento das matrizes:**

Gere os valores aleatórios para cada matriz com `rand()`.

4. **Exibição das matrizes:**

Use `loops for` para exibir os dados tabulados.

5. **Processamento dos dados:**

Faça os cálculos diretamente no programa. Use estruturas condicionais e acumuladores para calcular médias e encontrar máximos/mínimos.

6. **Exibição dos Relatórios:**

Imprima os resultados de forma clara e organizada no final do código.

Exemplo de Código Inicial (esqueleto):

```
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main() {
    const int NUM_BAIRROS = 5;
    const int NUM_DIAS = 7;

    int temperatura[NUM_BAIRROS][NUM_DIAS];
    int poluicao[NUM_BAIRROS][NUM_DIAS];
    int trafego[NUM_BAIRROS][NUM_DIAS];

    srand(time(0)); // Inicializa o gerador de números aleatórios

    // Preenchimento das matrizes e exibição inicial...
    // [Preencha de acordo com as tarefas descritas]

    return 0;
}
```

Boa Sorte!

Caso enfrente dificuldades, revise os conceitos apresentados em aula ou peça ajuda ao professor. O objetivo deste trabalho é consolidar seu aprendizado e prepará-lo para desafios mais avançados!