

QUESTÃO 1-

```
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>

using namespace std;

vector<vector<int>> lerVetores(const string& nomeArquivo) {
    ifstream arquivo(nomeArquivo);
    vector<vector<int>> vetores;
    string linha;

    while (getline(arquivo, linha)) {
        stringstream ss(linha);
        vector<int> vetor;
        int valor;
        while (ss >> valor) {
            vetor.push_back(valor);
        }
        vetores.push_back(vetor);
    }

    return vetores;
}

vector<int> somarVetores(const vector<vector<int>>& vetores) {
    if (vetores.empty()) return {};

    size_t maiorTamanho = 0;
    for (const auto& vetor : vetores) {
        if (vetor.size() > maiorTamanho) {
            maiorTamanho = vetor.size();
        }
    }

    vector<int> soma(maiorTamanho, 0);

    for (const auto& vetor : vetores) {
        for (size_t i = 0; i < vetor.size(); ++i) {
            soma[i] += vetor[i];
        }
    }

    return soma;
}

void salvarSoma(const string& nomeArquivo, const vector<int>& soma) {
```

```

    ofstream arquivo(nomeArquivo);
    for (const auto& valor : soma) {
        arquivo << valor << " ";
    }
    arquivo << endl;
}

int main() {
    string nomeArquivoEntrada = "vetores.txt";
    string nomeArquivoSaida = "soma.txt";

    vector<vector<int>> vetores = lerVetores(nomeArquivoEntrada);
    vector<int> soma = somarVetores(vetores);
    salvarSoma(nomeArquivoSaida, soma);

    cout << "A soma dos vetores foi salva em " << nomeArquivoSaida << endl;

    return 0;
}

```

QUESTÃO 2-

```

#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main() {
    string nomeArquivo = "mochileiro.txt";
    char caractere;
    cout << "Informe o caractere a ser contado: ";
    cin >> caractere;

    ifstream arquivo(nomeArquivo);
    if (!arquivo) {
        cerr << "Erro ao abrir o arquivo " << nomeArquivo << endl;
        return 1;
    }

    string linha;
    int numeroLinha = 1;

    while (getline(arquivo, linha)) {
        int contador = 0;
        for (char c : linha) {
            if (c == caractere) {
                contador++;
            }
        }
    }
}

```

```

    }
}
cout << "Linha " << numeroLinha << ": " << contador << " ocorrências do caractere '"
<< caractere << "'" << endl;
numeroLinha++;
}

arquivo.close();
return 0;
}

```

QUESTÃO 3-

```

#include <iostream>
#include <fstream>
#include <string>
#include <limits>

using namespace std;

// Função para ler dados de altura e peso
void lerDados(const string& nomeArquivo, string& codigoMaior, double& maiorValor) {
    ifstream arquivo(nomeArquivo);
    if (!arquivo) {
        cerr << "Erro ao abrir o arquivo " << nomeArquivo << endl;
        exit(1);
    }

    string codigo;
    double valor;
    maiorValor = numeric_limits<double>::lowest();

    while (arquivo >> codigo >> valor) {
        if (valor > maiorValor) {
            maiorValor = valor;
            codigoMaior = codigo;
        }
    }

    arquivo.close();
}

int main() {
    string codigoMaiorAltura, codigoMaiorPeso;
    double maiorAltura, maiorPeso;

    lerDados("a.txt", codigoMaiorAltura, maiorAltura);
    lerDados("peso.txt", codigoMaiorPeso, maiorPeso);
}

```

```

    cout << "Pessoa com maior altura: " << codigoMaiorAltura << " com " << maiorAltura << "
    metros" << endl;
    cout << "Pessoa com maior peso: " << codigoMaiorPeso << " com " << maiorPeso << "
    kg" << endl;

    return 0;
}

```

QUESTÃO 4-

```

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <limits>

using namespace std;

// Função para verificar se o aluno está cadastrado
bool alunoCadastrado(const string& nomeAluno) {
    ifstream arquivo(nomeAluno + ".txt");
    return arquivo.good();
}

// Função para cadastrar um novo aluno
void cadastrarAluno() {
    string nomeAluno, matricula;
    cout << "Informe o nome do aluno: ";
    cin >> nomeAluno;

    if (alunoCadastrado(nomeAluno)) {
        cout << "Aluno já cadastrado." << endl;
        return;
    }

    cout << "Informe a matrícula do aluno: ";
    cin >> matricula;

    ofstream arquivo(nomeAluno + ".txt");
    arquivo << matricula << endl;
    arquivo.close();
    cout << "Aluno cadastrado com sucesso." << endl;
}

// Função para adicionar nota a um aluno
void adicionarNota() {
    string nomeAluno;

```

```

double nota;
cout << "Informe o nome do aluno: ";
cin >> nomeAluno;

if (!alunoCadastrado(nomeAluno)) {
    cout << "Aluno não cadastrado." << endl;
    return;
}

cout << "Informe a nota do aluno: ";
cin >> nota;

ofstream arquivo(nomeAluno + ".txt", ios::app);
arquivo << nota << endl;
arquivo.close();
cout << "Nota adicionada com sucesso." << endl;
}

// Função para calcular a média das notas de um aluno
void calcularMedia() {
    string nomeAluno;
    cout << "Informe o nome do aluno: ";
    cin >> nomeAluno;

    if (!alunoCadastrado(nomeAluno)) {
        cout << "Aluno não cadastrado." << endl;
        return;
    }

    ifstream arquivo(nomeAluno + ".txt");
    string linha;
    getline(arquivo, linha); // Ignorar a matrícula

    vector<double> notas;
    while (getline(arquivo, linha)) {
        notas.push_back(stod(linha));
    }
    arquivo.close();

    if (notas.empty()) {
        cout << "Nenhuma nota registrada para o aluno." << endl;
        return;
    }

    double soma = 0;
    for (double nota : notas) {
        soma += nota;
    }
}

```

```
double media = soma / notas.size();  
cout << "A média do aluno " << nomeAluno << " é " << media << endl;  
}
```

// Função para exibir o menu e processar as opções

```
void exibirMenu() {  
    int opcao;  
    do {  
        cout << "\nMenu:\n";  
        cout << "1. Cadastrar aluno\n";  
        cout << "2. Adicionar nota de um aluno\n";  
        cout << "3. Calcular média de um aluno\n";  
        cout << "4. Fechar o programa\n";  
        cout << "Escolha uma opção: ";  
        cin >> opcao;  
  
        switch (opcao) {  
            case 1:  
                cadastrarAluno();  
                break;  
            case 2:  
                adicionarNota();  
                break;  
            case 3:  
                calcularMedia();  
                break;  
            case 4:  
                cout << "Programa encerrado." << endl;  
                break;  
            default:  
                cout << "Opção inválida. Tente novamente." << endl;  
        }  
    } while (opcao != 4);  
}
```

```
int main() {  
    exibirMenu();  
    return 0;  
}
```