**//Task 1**

#include <iostream>

#include <vector>

#include <climits>

using namespace std;

int summation(const vector<int>& arr) {

int sum = 0;

for (int num : arr) {

sum += num;

}

return sum;

}

int maximum(const vector<int>& arr) {

int max = arr.empty() ? INT\_MIN : arr[0];

for (int num : arr) {

if (num > max) max = num;

}

return max;

}

int main() {

int n;

cout << "Enter the number of elements: ";

cin >> n;

vector<int> arr(n);

cout << "Enter " << n << " integers:" << endl;

for (int i = 0; i < n; ++i) {

cin >> arr[i];

}

cout << "Summation: " << summation(arr) << endl;

cout << "Maximum: " << maximum(arr) << endl;

return 0;

}

**//Task 2**

#include <iostream>

#include <string>

#include <vector>

using namespace std;

struct Course {

string course\_code;

string course\_name;

};

struct Grade {

int mark;

char the\_grade;

void calculateGrade() {

if (mark > 69) the\_grade = 'A';

else if (mark > 59) the\_grade = 'B';

else if (mark > 49) the\_grade = 'C';

else if (mark > 39) the\_grade = 'D';

else the\_grade = 'E';

}

};

struct Student {

string registration\_number;

string name;

int age;

Course course;

Grade grade;

};

void addStudent(vector<Student>& students) {

if (students.size() >= 40) {

cout << "Maximum number of students reached." << endl;

return;

}

Student new\_student;

cout << "Enter registration number: ";

cin >> new\_student.registration\_number;

cout << "Enter name: ";

cin.ignore();

getline(cin, new\_student.name);

cout << "Enter age: ";

cin >> new\_student.age;

cout << "Enter course code: ";

cin >> new\_student.course.course\_code;

cout << "Enter course name: ";

cin.ignore();

getline(cin, new\_student.course.course\_name);

cout << "Enter mark: ";

cin >> new\_student.grade.mark;

new\_student.grade.calculateGrade();

students.push\_back(new\_student);

cout << "Student added successfully." << endl;

}

void editStudent(vector<Student>& students, const string& reg\_no) {

for (Student& student : students) {

if (student.registration\_number == reg\_no) {

cout << "Editing student: " << reg\_no << endl;

cout << "Enter new name: ";

cin.ignore();

getline(cin, student.name);

cout << "Enter new age: ";

cin >> student.age;

// Note: Course and grades are not edited for simplicity

cout << "Student details updated." << endl;

return;

}

}

cout << "Student with registration number " << reg\_no << " not found." << endl;

}

void displayStudents(const vector<Student>& students) {

if (students.empty()) {

cout << "No students to display." << endl;

return;

}

for (const Student& student : students) {

cout << "Registration Number: " << student.registration\_number << endl;

cout << "Name: " << student.name << endl;

cout << "Age: " << student.age << endl;

cout << "Course Code: " << student.course.course\_code << endl;

cout << "Course Name: " << student.course.course\_name << endl;

cout << "Mark: " << student.grade.mark << endl;

cout << "Grade: " << student.grade.the\_grade << endl;

cout << "---------------------------------" << endl;

}

}

int main() {

vector<Student> students;

int choice;

string reg\_no;

do {

cout << "\nStudent Management System" << endl;

cout << "1. Add Student" << endl;

cout << "2. Edit Student" << endl;

cout << "3. Display All Students" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

addStudent(students);

break;

case 2:

cout << "Enter Registration Number of the student to edit: ";

cin >> reg\_no;

editStudent(students, reg\_no);

break;

case 3:

displayStudents(students);

break;

case 4:

cout << "Exiting the system." << endl;

break;

default:

cout << "Invalid choice. Please try again." << endl;

}

} while (choice != 4);

return 0;

}

// task 4

