**TASK 1**

**1.CGPA Calculator**

**The user’s input is used to compute the CGPA, which includes information like the number of courses taken and the grades earned in each one. The program also shows each student’s overall course grade. CGPA Calculator calculates a student’s Cumulative Grade Point Average (CGPA) from the given exam results. The program can show the individual grades of each course, calculate total credits and total grade points achieved, determine the GPA for the semester, and based on all the data, it can generate and present the CGPA of the student**.

**CODE**

#include <iostream>

#include <vector>

using namespace std;

double calculateGPA(vector<double> grades, vector<int> credits)

{

double totalGradePoints = 0;

int totalCredits = 0;

for (size\_t i = 0; i < grades.size(); i++) {

totalGradePoints += grades[i] \* credits[i];

totalCredits += credits[i];

}

return totalCredits == 0 ? 0 : totalGradePoints / totalCredits;

}

int main()

{

int numCourses;

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

cin >> numCourses;

vector<double> grades(numCourses);

vector<int> credits(numCourses);

for (int i = 0; i < numCourses; i++)

{

cout << "Enter grade for course " << i + 1 << " (in GPA scale, e.g., 4.0, 3.7): ";

cin >> grades[i];

cout << "Enter credit hours for course " << i + 1 << ": ";

cin >> credits[i];

}

double semesterGPA = calculateGPA(grades, credits);

cout << "\nSemester GPA: " << semesterGPA << endl;

int totalPreviousCredits;

double previousCGPA;

cout << "Enter total previous credits (or 0 if none): ";

cin >> totalPreviousCredits;

cout << "Enter previous CGPA (or 0 if none): ";

cin >> previousCGPA;

double totalGradePoints = previousCGPA \* totalPreviousCredits + semesterGPA \* accumulate(credits.begin(), credits.end(), 0);

int totalCredits = totalPreviousCredits + accumulate(credits.begin(), credits.end(), 0);

double CGPA = totalCredits == 0 ? 0 : totalGradePoints / totalCredits;

cout << "\nCumulative GPA (CGPA): " << CGPA << endl;

return 0;

}

**OUTPUT**

