MTRE 2610 – Engineering algorithms and visualization – Dr. Kevin McFall

Homework – Structures

1. Complete implementations for the `student` member functions given the structure below in order to compute the GPAs for two students. The `credit` and `grade` data member arrays hold the number of credits for each course taken and its corresponding letter grade, respectively. The data member `numCourses` indicates how many of the 100 array elements contain valid data, i.e. the number of courses the student has completed.

The member function `setGrades` is passed arrays of length `numIn` containing the number of credits and letter grades for each of the `numIn` courses. The function should store this information in the data members.

The member function `addGrade` is passed a single course, with its corresponding number of credits and letter grade, to be appended to whatever information is already stored in the data members.

The member function `getGPA` will loop through all the course information stored in the data members and return the student's current GPA.

```
#include <iostream>
using namespace std;

struct student {
	int  credit[100], numCourses;
	char grade [100];
	void   setGrades(int creditIn[], char gradeIn[], int numIn);
	void   addGrade (int creditIn  , char gradeIn             );
	double getGPA();
};

int main() {
	const int numStart = 5;
	int  startCredit[numStart] = {  3 ,  4 ,  3 ,  2 ,  4  };
	char startGrade [numStart] = { 'A', 'A', 'B', 'C', 'B' };
	student me, you;
	me .setGrades(startCredit, startGrade, numStart);
	you.setGrades(startCredit, startGrade, numStart);
	me .addGrade(4, 'C'); me .addGrade(3, 'F');
	you.addGrade(2, 'B'); you.addGrade(4, 'A');
	cout << "  I have a " << me .getGPA() << " GPA" << endl;
	cout << "You have a " << you.getGPA() << " GPA" << endl;
	return 0;
}
```