Name: Ayush Bothra

Branch: Computer Engineering

Roll No: 231070011

SY Btech

DAA Assignment 1

Algorithm:

// calc\_SPI(sem, n):

Input: The first input is a 3d array that has in it the information of semesters needed, such that each array of semester in sem has a (credit, grade) pair. The second input is the number of semesters queried by the user.

Output: The function will output an array that will contain SPI for each semester. It will be of the same size as the second input n.

For i = 0 to n-1:

For j = 0 to size of each array in sem:

sum <- sum + sem[i][j][0]\*sem[i][j][1]

count <- count + sem[i][j][0]

end for

SPI\_store.append((sum/count))

end for

return SPI

calc\_CPI(sem, n)

Input: This function has the same input as the SPI function since it will call the function to obtain the SPI values.

Output: It will output a single CPI value.

For i = 0 to n-1:

CPI <- CPI + SPI\_store[i]

count <- count + 1

end for

CPI <- CPI / count

return CPI

Code: (file of code will also be uploaded)

#include<iostream>

#include<vector>

using namespace std;

vector<float> calc\_SPI(vector<vector<vector<int>>> sem,int n)

{

    vector<float> SPI\_store;

    if(n > sem.size()) return {-1}; // more sems queried than given.

    if(sem.size() == 0) return {}; // if no sems are provided

    for(int i = 0; i < n; i++)

    {

        float sum = 0; float count = 0;

        for(int j = 0; j < sem[i].size(); j++)

        {

            if(sem[i][j][1] > 10 || sem[i][j][0] > 4) return {-1}; // if grade scores are greater than 10 points or credits greater than 4

            if(sem[i][j][1] < 0 || sem[i][j][0] < 0) return {-1}; // if grade or credit scores are negative

            sum += sem[i][j][0] \* sem[i][j][1];

            count += sem[i][j][0];

        }

        SPI\_store.push\_back((sum/count));

    }

    return SPI\_store;

}

float calc\_CPI(vector<vector<vector<int>>> sem,int n)

{

    vector<float> values = calc\_SPI(sem,n);

    if(values.size() == 0) return 0;

    if(values[0] == -1) return -1;

    float sum = 0; float count = 0;

    for(const auto& value : values) // goes thru each value present in the array

    {

        sum += value;

        count = count + 1.0;

    }

    float CPI\_value = sum/count;

    return CPI\_value;

}

int main()

{

    vector<vector<vector<int>>> store = {{{2,9},{4,7},{2,8}},{{3,7},{1,10},{2,5}}};

    int n = 2;

    float cpi = calc\_CPI(store, n);

    cout << cpi << "\n";

    return 0;

}

Test Cases:

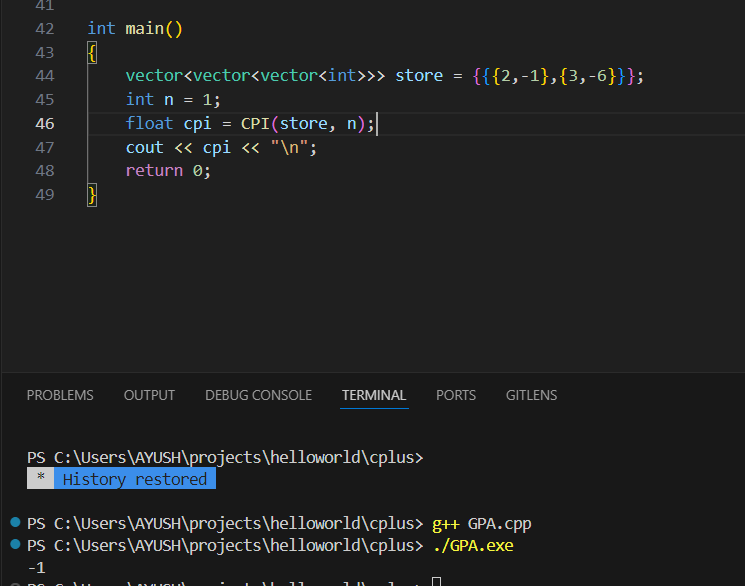
Case 1: sem = {}, n = 0

Since there are no semesters for calculation, the program will return 0 as the ans.



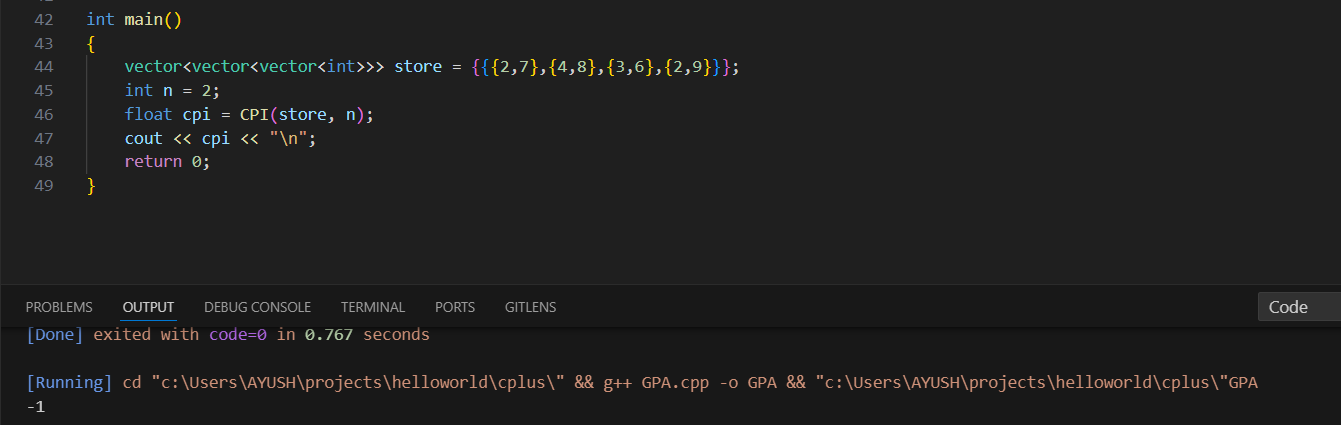
Case 2: sem = {{{2,-1},{3,-6}}}, n = 1

Since the grades are negative, the program will return -1 as indicator of error.



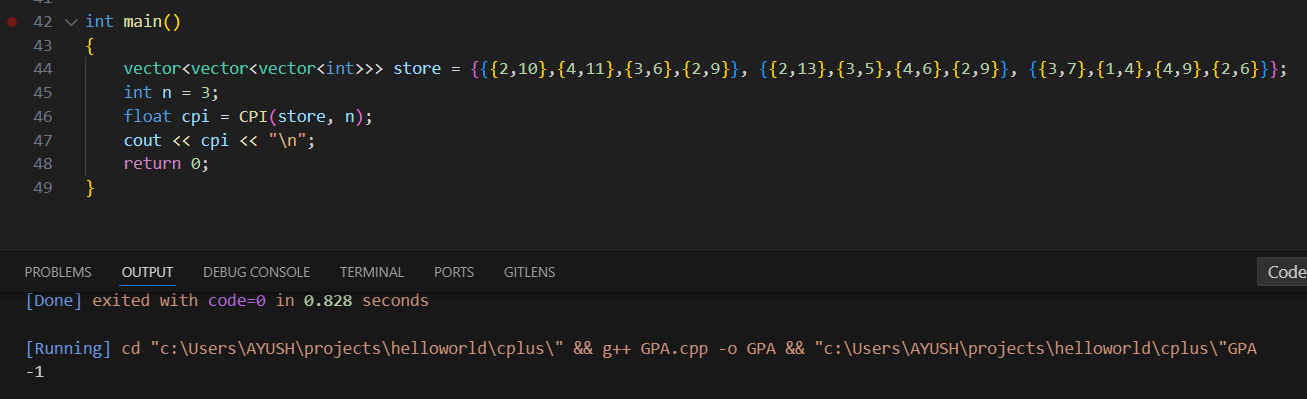
Case 3: sem = {{{2,7}, {4,8}, {3,6}, {2,9}}}, n = 2

The number of semester query is larger than the number of semesters given, so the program will return -1 as sign of error.



Case 4: sem = {{{2,10},{4,11},{3,6},{2,9}}, {{2,13},{3,5},{4,6},{2,9}}, {{3,7},{1,4},{4,9},{2,6}}}, n = 3

Since the grades are greater than 10 in the sems given in the vector, the program will output -1 as sign of error.



Case 5: sem = {{{2,9},{4,7},{2,8}},{{3,7},{1,10},{2,5}}}, n = 2

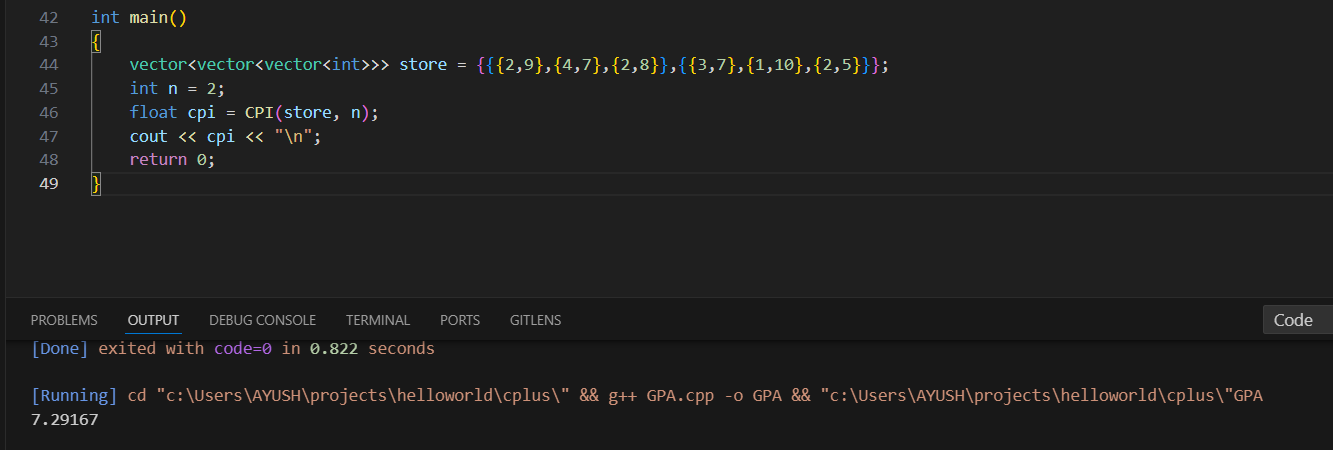
In the SPI function: (2\*9 + 4\*7 + 2\*8) / (2 + 4 +2) = 62 / 8 = 7.75 for the first sem.

The array now contains = [7.75]

(3\*7 + 1\*10 + 2\*5) / (3 + 1 + 2) = 41 / 6 = 6.8333

The array now contains = [7.75,6.8333]

In the CPI function: (7.75 + 6.8333) / (1+1) = (14.5833) / 2 = 7.29167 becomes the answer.



Conclusion: Thus, we have developed an algorithm that finds the CPI and SPI of a student.