1. We want to calculate the total marks of each student of a class in Physics, Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subject of each student Roll number of each student will be generated automatically.

#include <iostream>

#include <vector>

using namespace std;

class Marks {

public:

int rollNo;

string name;

Marks(string name) : name(name), rollNo(static\_cast<int>(nextRollNo++)) {}

private:

static int nextRollNo;

};

int Marks::nextRollNo = 1;

class Physics : public Marks {

public:

int marks;

Physics(string name, int marks) : Marks(name), marks(marks) {}

};

class Chemistry : public Marks {

public:

int marks;

Chemistry(string name, int marks) : Marks(name), marks(marks) {}

};

class Mathematics : public Marks {

public:

int marks;

Mathematics(string name, int marks) : Marks(name), marks(marks) {}

};

int main() {

int numStudents;

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

cin >> numStudents;

vector<Physics> physicsStudents;

vector<Chemistry> chemistryStudents;

vector<Mathematics> mathStudents;

string name;

int physicsMarks, chemistryMarks, mathMarks;

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

cout << "Enter name of student " << i + 1 << ": ";

cin >> name;

cout << "Enter Physics marks: ";

cin >> physicsMarks;

cout << "Enter Chemistry marks: ";

cin >> chemistryMarks;

cout << "Enter Mathematics marks: ";

cin >> mathMarks;

physicsStudents.push\_back(Physics(name, physicsMarks));

chemistryStudents.push\_back(Chemistry(name, chemistryMarks));

mathStudents.push\_back(Mathematics(name, mathMarks));

}

// Calculate total and average marks

int totalPhysics = 0, totalChemistry = 0, totalMath = 0;

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

totalPhysics += physicsStudents[i].marks;

totalChemistry += chemistryStudents[i].marks;

totalMath += mathStudents[i].marks;

}

double avgPhysics = static\_cast<double>(totalPhysics) / numStudents;

double avgChemistry = static\_cast<double>(totalChemistry) / numStudents;

double avgMath = static\_cast<double>(totalMath) / numStudents;

// Output results

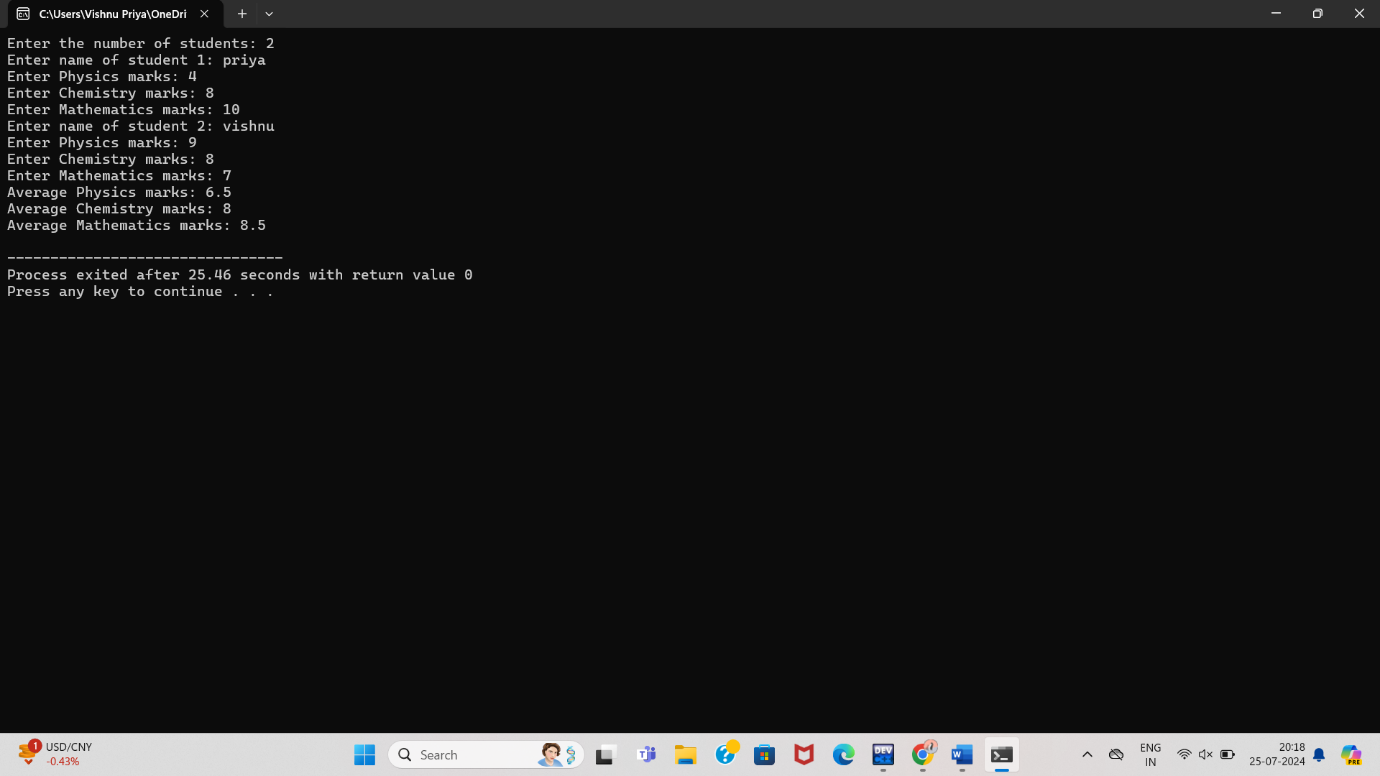
cout << "Average Physics marks: " << avgPhysics << endl;

cout << "Average Chemistry marks: " << avgChemistry << endl;

cout << "Average Mathematics marks: " << avgMath << endl;

return 0;

}



1. Given an integer atrray A[ ]consisting of N non-negative integers representing an elevation map, Where the width of each bar is 1. The task is to compute the total volume of water that can be trapped after rain.

#include <iostream>

#include <vector>

using namespace std;

int trap(vector<int>& height) {

int n = height.size();

int left = 0, right = n - 1;

int left\_max = 0, right\_max = 0;

int water = 0;

while (left < right) {

if (height[left] < height[right]) {

left\_max = max(left\_max, height[left]);

water += left\_max - height[left];

left++;

} else {

right\_max = max(right\_max, height[right]);

water += right\_max - height[right];

right--;

}

}

return water;

}

int main()

{

int n;

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

cin >> n;

vector<int> height(n);

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

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

cin >> height[i];

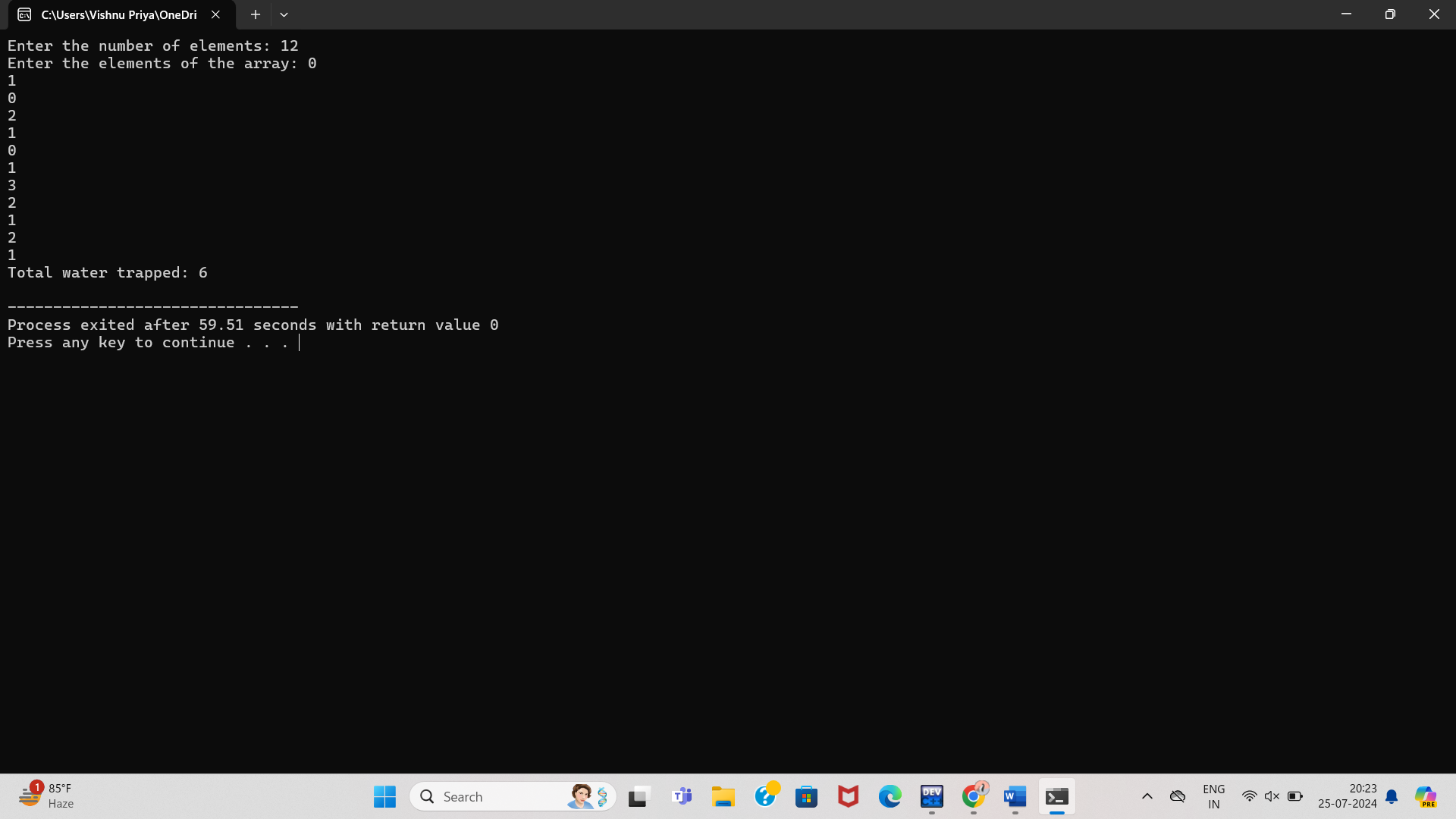
}

int trapped\_water = trap(height);

cout << "Total water trapped: " << trapped\_water << endl;

return 0;

}



1. Define a class Ele\_Bill in C++ with the following descriptions: Private Members: Cname of type character array Pnumber of type long No\_of \_Units of type integer Amount of type float. Calc\_amount() this number function should calculate the amount as No\_of\_Units.

#include <iostream>

#include <cstring>

using namespace std;

class Ele\_Bill {

private:

char Cname[50];

long Pnumber;

int No\_of\_Units;

float Amount;

void Calc\_amount() {

Amount = No\_of\_Units;

}

public:

Ele\_Bill(const char\* cname, long pnumber, int units) {

strcpy(Cname, cname);

Pnumber=pnumber;

No\_of\_Units = units;

Calc\_amount();

}

void displayBill() {

cout << "Customer Name: " << Cname << endl;

cout << "Phone Number: " << Pnumber << endl;

cout << "Number of Units: " << No\_of\_Units << endl;

cout << "Amount: Rs." << Amount << endl;

}

};

int main() {

char name[50];

long pnum;

int units;

cout << "Enter customer name: ";

cin.getline(name, 50);

cout << "Enter phone number: ";

cin>>pnum;

cout << "Enter number of units consumed: ";

cin >> units;

Ele\_Bill bill(name, pnum, units);

bill.displayBill();

return 0;

}

