**Muhammad Rayyan**

**Section: B**

**456847**

**Lab Manual FOP 10**

#include<bits/stdc++.h>

using namespace std;

**// Q: 01**

/\* int main() {

vector<int> myVector;

for (int i = 1; i <= 10; ++i) {

myVector.push\_back(i);

}

cout << "The elements in the vector are : ";

for (auto it = myVector.begin(); it != myVector.end(); ++it) {

cout << \*it << " "; // \*it is a pointer

}

cout <<endl;

myVector.push\_back(5); // already defined function of push back in standard library.

if (myVector.size() > 5) {

myVector.erase(myVector.begin() + 5);

}

cout << "Vector element after pushing 5 and removing the element at position 5 : ";

for (const auto& element : myVector) {

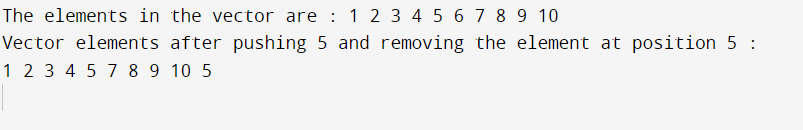
cout << element << " ";

}

cout << endl;

return 0;

} \*/



**//Q: 02**

/\* double calculate\_mean(const vector<int>& grades) {

int sum = 0;

for (int grade : grades) {

sum += grade;

}

return static\_cast<double>(sum) / grades.size();

}

double calculate\_median(const vector<int>& grades) {

vector<int> sortedGrades = grades;

sort(sortedGrades.begin(), sortedGrades.end());

size\_t size = sortedGrades.size();

if (size % 2 == 0) {

return (sortedGrades[size / 2 - 1] + sortedGrades[size / 2]) / 2.0;

} else {

return sortedGrades[size / 2];

}

}

vector<int> calculate\_mode(const vector<int>& grades) {

map<int, int> gradeCount;

for (int grade : grades) {

gradeCount[grade]++;

}

int maxFrequency = 0;

for (const auto& pair : gradeCount) {

maxFrequency = max(maxFrequency, pair.second);

}

vector<int> modeGrades;

for (const auto& pair : gradeCount) {

if (pair.second == maxFrequency) {

modeGrades.push\_back(pair.first);

}

}

return modeGrades;

}

int main() {

vector<string> names;

vector<int> grades;

int numPairs;

cout << "Enter the number of name/grade pairs: ";

cin >> numPairs;

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

string name;

int grade;

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

cin >> name;

cout << "Enter grade " << i + 1 << " : ";

cin >> grade;

names.push\_back(name);

grades.push\_back(grade);

}

double mean = calculate\_mean(grades);

cout << "The Mean of grades is : " << mean << std::endl;

double median = calculate\_median(grades);

cout << "The Median of grades is : " << median << std::endl;

vector<int> modeGrades = calculate\_mode(grades);

cout << "The Mode of grades is : ";

for (int mode : modeGrades) {

cout << mode << " ";

}

cout << endl;

cout << "The names a/c to their grades: ";

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

if (find(modeGrades.begin(), modeGrades.end(), grades[i]) != modeGrades.end()) {

cout << names[i] << " ";

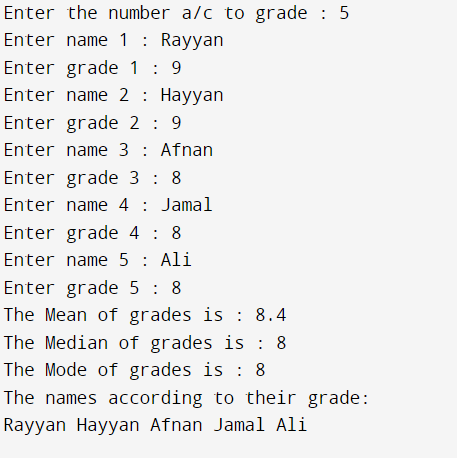
}

}

cout << std::endl;

return 0;

} \*/



**// Q: 03**

/\* class Triangle {

private:

double side1, side2, side3;

public:

Triangle(double s1, double s2, double s3) : side1(s1), side2(s2), side3(s3) {}

double calculatePerimeter() const {

return side1 + side2 + side3;

}

double calculateArea() const {

return sqrt(s \* (s - side1) \* (s - side2) \* (s - side3));

}

void printDetails() const {

cout << "The 3 triangle sides are : " << side1 << " m, " << side2 << " m, " << side3 << " m" << std::endl;

cout << "Perimeter of traingle is : " << calculatePerimeter() << " m" << std::endl;

cout << "Area of triangle is : " << calculateArea() << " meter square" << std::endl;

}

} ;

int main() {

Triangle myTriangle(3.0, 4.0, 5.0);

myTriangle.printDetails();

return 0;

} \*/

